Towards Scalable, Flexible, and Interpretable Self-Supervised Learning for Multiview Biomedical Data by James Chapman

December 2023

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

Imagine a world where images of you, data from your smartwatch, and your electronic health records could seamlessly integrate to paint a comprehensive picture of your health. Now, envision this on a global scale, where vast amounts of diverse biomedical data are harnessed to target drug trials, personalize treatment, and improve the lives of millions. This is the promise of multiview learning in the era of big data, but it comes with a significant challenge: How can we effectively integrate and analyze these complex, heterogeneous data sources?

The need for novel methods to tackle this challenge is paramount. Traditional approaches often struggle with the sheer scale and intricacy of modern biomedical datasets, limiting our ability to uncover crucial insights and advance personalized medicine. This thesis addresses this critical need by developing cutting-edge machine learning techniques that leverage the power of self-supervised and multiview learning, focusing on improving Canonical Correlation Analysis (CCA) for enormous datasets with rich structure and complex, possibly non-linear relationships.

The primary contributions of this thesis are fourfold. First, a framework for regularised CCA using structured priors is developed, enhancing the interpretability of the results. Second, simulated data generation methods for CCA are unified under a latent variable model perspective, improving our understanding of the relationship between loadings and weights in CCA. Third, a new gradient descent approach for CCA and other generalised eigenvalue problems is formulated, tailored for large datasets. Finally, this gradient descent approach is extended to Deep CCA and Joint Embedding Self-Supervised Learning, enabling the integration of diverse data sources using modern deep learning techniques. Finally, we make all of our code and data publicly available, ensuring that our research is reproducible and accessible to the wider scientific community.

Impact Statement

The research presented in this thesis has the potential to significantly advance the field of representation learning, particularly in the context of integrating diverse, large-scale biomedical data. By developing novel methods for canonical correlation analysis (CCA) and its extensions, this work addresses the critical challenge of uncovering meaningful patterns and relationships in complex, high-dimensional datasets.

Within academia, the theoretical contributions of this thesis will enable researchers to scale CCA methods to much larger datasets, a crucial development as access to extensive biomedical data becomes increasingly common. This will facilitate the discovery of new insights and knowledge across various domains, from neuroscience to genomics, ultimately leading to a deeper understanding of human health and disease.

Beyond the academic sphere, the impact of this research extends to numerous real-world applications. The high-quality, open-source implementations of several CCA methods developed as part of this thesis will promote reproducible research and widespread adoption by the Python community, which has become the de facto standard for data science and machine learning. By providing accessible tools and frameworks, this work democratizes the use of advanced representation learning techniques, allowing practitioners and researchers from diverse backgrounds to harness the power of these methods in their own domains.

Through this mechanism, the work presented in this thesis has already demonstrated impact in fields as varied as process monitoring, geothermal flow, and medical imaging. As more researchers and practitioners adopt these tools and techniques, we anticipate far-reaching implications for industries such as healthcare, where improved integration and analysis of biomedical data could lead to earlier disease detection, personalized treatment plans, and enhanced patient outcomes.

In summary, by pushing the boundaries of representation learning and providing

practical, open-source tools for the research community, this thesis has the potential to accelerate discovery and innovation across a wide range of domains, with particularly profound implications for advancing our understanding of human health and well-being.

Acknowledgements

Thanks to my supervisors, Professor Janaina Mourao-Miranda and Professor John Shawe-Taylor, for their contributions. I am very grateful to the EPSRC UCL Centre for Doctoral Training (CDT) in Intelligent, Integrated Imaging in Healthcare (i4Health) and NIHR UCLH Biomedical Research Centre for funding this research. Thanks to G-Research for funding my trip to NeurIPS 2023 to present my work.

Cemre, we could and should have done so much more together, but I am grateful for your advice before you left. Agoston, I was inspired your immense knowledge of the field.

Ana, you kept our paper alive when I had given up on it[1].

Lennie, you told me that my work was[2] rubbish, and you made it better.

The friends from 90 High Holborn including the Mojo Dojo Casa House who made my NeurIPS 2023 experience unforgettable. Florence, I was so honoured to be included on Fusili and the hard lessons you taught me in marketing. Deji, I wish I could have some of your relentless optimism.

The boat clubs of University College London, University of London, and Vesta Rowing Club that have given me an outlet and goals that have given me a sense of progress, purpose, and community even as academia has sometimes got me[3] down.

Muvs and Farvs, obviously this has been a bit of a disaster but we got there.

Rebecca, ****ing hell. We got through this together and

[1] also the PhD more generally
[2] still
[3] very

List of Publications

First Author Peer Reviewed Conference Proceedings

Chapman, James, Lennie Wells, and Ana Lawry Aguila (2024). *Unconstrained Stochastic CCA: Unifying Multiview and Self-Supervised Learning*.

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 (2023). *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

Adams, Rick A. et al. (2024). "Voxel-wise multivariate analysis of brain-psychosocial associations in adolescents reveals six latent dimensions of cognition and psychopathology". In: *Biological Psychiatry: Cognitive Neuroscience and Neuroimag-*

ing. ISSN: 2451-9022. DOI: https://doi.org/10.1016/j.bpsc.2024.03.006. URL: https://www.sciencedirect.com/science/article/pii/S2451902224000855.

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*.

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.

Software

Mihalik, Agoston, Nils Winter, et al. (Oct. 2022). *CCA/PLS Toolkit*. Version 1.0.0. URL: https://github.com/anaston/cca_pls_toolkit.

Townsend, Florence, James Chapman, and James Cole (Nov. 2023). *florencejt/fusilli: Fusilli v1.0.0*. Version v1.0.0. DOI: 10.5281/zenodo.10228564. URL: https://doi.org/10.5281/zenodo.10228564.

UCL Research Paper Declaration Form: referencing the doctoral candidate's own published work(s)

1. **1. For a research manuscript that has already been published** (if not yet published, please skip to section 2)**:**
   (a) **What is the title of the manuscript?** CCA with Shared Weights for Self-Supervised Learning
   (b) **Please include a link to or doi for the work:**
   (c) **Where was the work published?** 4th Workshop on Self-Supervised Learning: Theory and Practice
   (d) **Who published the work?**
   (e) **When was the work published?** December 2023
   (f) **List the manuscript's authors in the order they appear on the publication:** James Chapman and Lennie Wells
   (g) **Was the work peer reviewd?** Yes
   (h) **Have you retained the copyright?** Yes
   (i) **Was an earlier form of the manuscript uploaded to a preprint server (e.g. medRxiv)? If 'Yes', please give a link or doi**
   If 'No', please seek permission from the relevant publisher and check the box next to the below statement:
   ☒ *I acknowledge permission of the publisher named under 1d to include in this thesis portions of the publication named as included in 1c.*

2. **For a research manuscript prepared for publication but that has not yet been published** (if already published, please skip to section 3)**:**
   (a) **What is the current title of the manuscript?**
   (b) **Has the manuscript been uploaded to a preprint server 'e.g. medRxiv'? If 'Yes', please please give a link or doi:**
   (c) **Where is the work intended to be published?**
   (d) **List the manuscript's authors in the intended authorship order:**
   (e) **Stage of publication:**

3. **For multi-authored work, please give a statement of contribution covering all authors** (if single-author, please skip to section 4)**:**

4. **In which chapter(s) of your thesis can this material be found?**

e-Signatures confirming that the information above is accurate (this form should be co-signed by the supervisor/ senior author unless this is not appropriate, e.g. if the paper was a single-author work)**:**
James Chapman
Candidate: James Chapman
Date: 17/04/2024

Supervisor/Senior Author signature (where appropriate)**:**
Date:

UCL Research Paper Declaration Form: referencing the doctoral candidate's own published work(s)

1. **1. For a research manuscript that has already been published** (if not yet published, please skip to section 2)**:**
   (a) **What is the title of the manuscript?** Unconstrained Stochastic CCA: Unifying Multiview and Self-Supervised Learning
   (b) **Please include a link to or doi for the work:** https://iclr.cc/virtual/2024/poster/18714
   (c) **Where was the work published?** Proceedings of the International Conference on Learning Representations
   (d) **Who published the work?** International Conference on Learning Representations
   (e) **When was the work published?** May 2024
   (f) **List the manuscript's authors in the order they appear on the publication:** James Chapman and Lennie Wells and Ana Lawry Aguila
   (g) **Was the work peer reviewd?** Yes
   (h) **Have you retained the copyright?** Yes
   (i) **Was an earlier form of the manuscript uploaded to a preprint server (e.g. medRxiv)? If 'Yes', please give a link or doi** https://arxiv.org/abs/2310.01012
   If 'No', please seek permission from the relevant publisher and check the box next to the below statement:
      ☒ *I acknowledge permission of the publisher named under 1d to include in this thesis portions of the publication named as included in 1c.*

2. **For a research manuscript prepared for publication but that has not yet been published** (if already published, please skip to section 3)**:**
   (a) **What is the current title of the manuscript?**
   (b) **Has the manuscript been uploaded to a preprint server 'e.g. medRxiv'? If 'Yes', please please give a link or doi:**
   (c) **Where is the work intended to be published?**
   (d) **List the manuscript's authors in the intended authorship order:**
   (e) **Stage of publication:**

3. **For multi-authored work, please give a statement of contribution covering all authors** (if single-author, please skip to section 4)**:**
4. **In which chapter(s) of your thesis can this material be found?** 5,6

e-Signatures confirming that the information above is accurate (this form should be co-signed by the supervisor/ senior author unless this is not appropriate, e.g. if the paper was a single-author work)**:**
James Chapman
Candidate: James Chapman
Date: 17/04/2024

Supervisor/Senior Author signature (where appropriate)**:**
Date:

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Acronyms

ADNI Alzheimer's Disease Neuroimaging Initiative. 13, 58–60, 71, 73, 74, 78, 79, 81, 188

CCA Canonical Correlation Analysis. 31, 44–47, 49, 50, 55, 56, 61, 94–96, 98, 99, 102, 108–110, 123, 125

DCCA Deep Canonical Correlation Analysis. 46

FRALS Flexible Regularised Alternating Least Squares. 70, 71, 74

GFA Group Factor Analysis. 98, 99

HCP Human Connectome Project. 13, 58–60, 71–75, 78–81, 186

KCCA Kernel Canonical Correlation Analysis. 125

MCCA Multiset Canonical Correlation Analysis. 47

MRI Magnetic Resonance Imaging. 69

PCA Principal Component Analysis. 39–42, 44, 123–125

PLS Partial Least Squares. 42–47, 108–110, 123

Symbols List

$W^{(i)}$ The matrix of loadings for the i-th view. The jk-th element of this matrix is given by $w_{jk}^{(i)}$..

$X^{(i)}$ The ith view of the data, represented as a matrix of random variables. $X^{(i)} \in \mathbb{R}^{D_i}$ where D_i is the dimensionality of the ith view..

$Z^{(i)}$ The learned K-dimensional representation for the ith view of the data. $Z^{(i)} = f^{(i)}(X^{(i)}; \theta^{(i)})$ where $f^{(i)}$ is a function parameterized by $\theta^{(i)}$..

$\mathrm{CCA}_K(X^{(1)}, X^{(2)})$ The vector of the top K canonical correlations obtained from Canonical Correlation Analysis (CCA) applied to views $X^{(1)}$ and $X^{(2)}$. It is defined as $\mathrm{CCA}_K(X^{(1)}, X^{(2)}) := (\rho_k)_{k=1}^K$, where ρ_k is the kth canonical correlation..

$\mathrm{MCCA}_K(X^{(1)}, \ldots, X^{(I)})$ The vector of the top K generalized eigenvalues obtained from Multiview Canonical Correlation Analysis (MCCA) applied to views $X^{(1)}, \ldots, X^{(I)}$. It is defined as $\mathrm{MCCA}_K(X^{(1)}, \ldots, X^{(I)}) = (\lambda_1, \ldots, \lambda_K)$, where λ_k is the kth generalized eigenvalue..

$\mathrm{PLS}_K(X^{(1)}, X^{(2)})$ The vector of the top K singular values obtained from Partial Least Squares (PLS) applied to views $X^{(1)}$ and $X^{(2)}$. It represents the covariances between the learned latent variables..

Σ_{ii} The population covariance matrix of the random variables associated with view i. $\Sigma_{ii} = \mathrm{Cov}(X^{(i)})$..

Σ_{ij} The population cross-covariance matrix between the random variables associated with view i and view j. $\Sigma_{ij} = \mathrm{Cov}(X^{(i)}, X^{(j)})$..

λ The generalized eigenvalues obtained when solving the generalized eigenvalue problems that arise in PLS and CCA. In CCA, these are known as the canonical correlations..

ρ_k The kth canonical correlation obtained from CCA. ρ_k is the kth element of the vector $\mathrm{CCA}_K(X^{(1)}, X^{(2)})$..

$\theta^{(i)}$ The parameters of the function $f^{(i)}$ used to learn the representation $Z^{(i)}$ from the ith view $X^{(i)}$..

$u_j^{(i)}$ The weight of the j-th feature in the i-th view for a latent variable..

$w_j^{(i)}$ The loading of the j-th feature in the i-th view on the k-th latent variable..

Definitions

$U^{(i)}$ The matrix of weights for the i-th view. The jk-th element of this matrix is given by $u_{jk}^{(i)}$..

canonical correlations In the context of CCA, the generalized eigenvalues λ are referred to as canonical correlations. They represent the strength of the linear relationship between the learned representations of the two views. The goal of CCA is to find the weights that maximize these canonical correlations.

covariance matrix A covariance matrix captures the relationships between variables in a dataset. The notation Σ_{ij} represents the population covariance matrix between the variables in view i and view j. Each element of this matrix, $\Sigma_{ij}(a, b)$, measures how much the a-th variable in view i and the b-th variable in view j change together. A positive covariance indicates that the variables tend to increase or decrease together, while a negative covariance indicates that they tend to move in opposite directions. Σ_{ii} represents the covariance matrix within view i, capturing the relationships between variables in the same view. These matrices are essential for understanding the structure of the data and are used in subspace learning algorithms like CCA and PLS to find common patterns across views. , 38, 40, 41, 46–49

generalized eigenvalue problem A Generalized Eigenvalue Problem (GEP) is defined by two symmetric matrices $A, B \in \mathbb{R}^{D \times D}$ and is characterized by the set of solutions to the equation $Au = \lambda Bu$, where $\lambda \in \mathbb{R}$ and $u \in \mathbb{R}^D$ are called generalized eigenvalue and generalized eigenvector, respectively. Many classical subspace learning algorithms, including CCA and PLS, can be formulated as GEPs constructed from covariance matrices.

latent variables Latent variables, also referred to as representations, are the low-dimensional variables Z_k computed as linear transformations of the input

variables using the weights, i.e., $Z_k = X_k u_k$. They aim to capture meaningful structures in the data that are not directly observed. , 38

loadings The Pearson correlation between a feature and a latent variable, given by $\mathrm{Corr}(X_j^{(i)}, Z_k)$. It measures the strength of the linear relationship between a feature and a latent variable, with higher absolute values indicating a stronger relationship. Loadings are invariant to certain transformations of the data matrix, such as scaling, duplication, and linear combinations of columns.. 17, 19, 38, 108, 109, 112–117, 119, 120, 187, 189

norm A norm is a function that assigns a non-negative length or size to a vector in a vector space. Common norms include the L1 norm (sum of absolute values) and the L2 norm (Euclidean norm)..

representations The representations or latent variables Z_k are computed as linear transformations of the input variables using the weights, i.e., $Z_k = X_k u_k$. They aim to capture meaningful low-dimensional structures in the data. In the CCA literature, they are sometimes referred to as canonical variables. 16, 32–34, 37–40, 42, 78, 80, 88

sample covariance matrix In practice, we often don't have access to the true population covariance matrices, which would require knowing the data distribution. Instead, we estimate these matrices from the available data samples. The sample covariance matrix, denoted as $\hat{\Sigma}^{(ij)}$, is calculated by averaging the products of the centered data points (i.e., data points with the mean subtracted) across all samples. These sample covariance matrices serve as approximations of the true population covariance matrices and are used in place of them when applying subspace learning algorithms like CCA and PLS to real-world datasets. , 50

views Views refer to the different sets of variables or modalities that describe the same underlying phenomena or objects. In the context of multiview learning, methods like PLS and CCA aim to find common latent structures that explain the relationships between these views. The term "view" is used to emphasize the distinct nature of these variable sets, which can come from different data sources or represent different aspects of the data. , 32–34, 36, 41, 42, 44, 47, 51, 56

weights When the functions f are linear, the weights u_k are used to compute the representations or latent variables as $Z_k = X_k u_k$. The weights define the linear transformation from the input variables to the latent space. 16, 19, 20, 38, 44, 47, 61, 62, 68, 77–80, 86, 88, 120, 189

Chapter I

Introduction

It was June 2021, and I had self-referred to the Community Living Well service in London, UK, seeking help for my mental health. Each week, I met with my therapist and dutifully filled out the questionnaires, rating my mood and answering questions about my well-being. Yet, I couldn't shake the feeling that these snapshots were inadequate in capturing the complexity of my mental state. As a keen sportsperson, I relied on my Garmin watch to track my heart rate, sleep, and activity levels, providing a continuous stream of biometric data that painted a more nuanced picture of my physical health. Moreover, as a type 1 diabetic, my continuous glucose monitor offered real-time insights into my blood sugar levels, helping me fine-tune my insulin management. These diverse data streams, each offering unique perspectives on my overall health, highlight the potential of learning meaningful representations from disparate data sources to gain a more comprehensive understanding of an individual's well-being.

In biomedical research, there is a growing need to develop methods that can effectively combine and analyze data from various sources, such as electronic health records, imaging data, and patient-reported outcomes. By leveraging the power of self-supervised learning, a machine learning approach that learns from unlabeled data by predicting parts of the input from other parts, we can potentially uncover hidden patterns and relationships in these complex datasets. Self-supervised learning is particularly well-suited for this task, as it can learn robust and generalizable representations from vast amounts of unlabeled data, which is abundant in the biomedical domain.

This thesis focuses on developing and applying novel machine learning methods to address the challenge of integrating diverse health metrics through representation

learning. A key approach explored in this work is Canonical Correlation Analysis (CCA), a powerful multiview learning technique that aims to find linear transformations of two or more datasets such that the transformed variables are maximally correlated. By learning these transformations, CCA can uncover latent structures and relationships between disparate data sources, making it a valuable tool for representation learning in the biomedical domain. Through improved methods for multiview and self-supervised learning, particularly centered around CCA, we hope to improve the analysis and comprehension of biomedical data, ultimately enhancing our ability to understand and manage personal health.

The main contributions of this thesis are fourfold:

1. Developing a framework for regularized Canonical Correlation Analysis (CCA) using structured priors to learn more interpretable and biologically meaningful representations;
2. Unifying simulated data generation methods for CCA under a latent variable model perspective to facilitate the evaluation and comparison of representation learning algorithms;
3. Formulating a new gradient descent approach for CCA and other generalized eigenvalue problems, tailored for learning representations from large datasets;
4. Extending the gradient descent approach to Deep CCA and Joint Embedding Self-Supervised Learning to learn more complex representations from complex, high-dimensional data;
5. Developing CCA-Zoo, an open-source Python package for Canonical Correlation Analysis, which provides a unified interface for various CCA methods and facilitates their application in representation learning.

These contributions have significant practical implications, from aiding in the diagnosis and treatment of neurological disorders to enabling efficient analysis of extensive health databases like the UK Biobank.

1 Thesis Structure

The thesis is structured as follows:

- **Chapter II** reviews multiview and self-supervised learning techniques, focusing on their application in learning meaningful representations from 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 to showcase the potential of learning structured representations.
- **Chapter IV** examines the relationship between loadings and weights in CCA, using simulated data to show the advantages of loadings for interpreting learned representations.
- **Chapter V** presents a new gradient descent algorithm for generalized eigenvalue problems, tailored for learning representations from large datasets, demonstrated with Multiview CCA and PLS. We show how our algorithm can be applied to large datasets, using the UK Biobank as an example.
- **Chapter VI** extends the algorithm from Chapter V to deep learning, showing its application in scaling deep CCA to learn hierarchical representations from complex, high-dimensional data. We demonstrate state-of-the-art results on CIFAR-10 and CIFAR-100 benchmarks, illustrating the potential of Deep CCA in Self-Supervised Learning.
- **Chapter VII** introduces CCA-Zoo, a Python package implementing the methodologies of this thesis, and discusses its role in the Python ecosystem and biomedical research, particularly in facilitating representation learning.
- **Chapter VIII** discusses the implications, challenges, and future directions for the research presented in this thesis.

Through this thesis, we aspire to bridge the gap between the potential of biomedical data and the current capabilities of analytical methods. By developing novel, scalable, and interpretable machine learning approaches for representation learning, we aim to unlock the full potential of diverse health metrics, paving the way for advancements in biomedical research and personalized healthcare.

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.
>
> Professor David MacKay

Contents

1 Introduction

This chapter provides the foundational knowledge needed to understand the thesis as a whole, while the individual chapters will provide more specific background information as needed.

2 Machine Learning and Multiview Learning

Machine learning encompasses methods that enable models to learn patterns and make decisions from data. Machine learning methods are typically categorized by a training set of data, which is used to learn a model, and a test set of data, which is used to evaluate the model.

Arguably the most common machine learning paradigm is supervised learning, where the training data consists of pairs of inputs and outputs, and the model learns to predict a function that maps the inputs to the outputs. This function is then used to predict the outputs for new inputs. The goal of supervised learning is to learn a function that generalizes well to new data, i.e., to make accurate predictions on

unseen data. At the heart of many machine learning algorithms lies the concept of representation learning - the process of automatically discovering and extracting meaningful features or representations from raw data. Representation learning aims to transform high-dimensional, complex data into a lower-dimensional space that captures the most salient aspects of the original data. This process not only reduces computational complexity but also uncovers hidden structures and relationships within the data, which can then be leveraged for various tasks such as classification, regression, or clustering.

Unsupervised and self-supervised learning are common machine learning paradigms that often involve learning low-dimensional *latent* representations of the input data. In these paradigms, the training data consists of inputs only, and the model learns to find patterns or structure in the data without explicit output labels. *Downstream* models can use these learned representations as their inputs rather than the original data.

While the distinction between unsupervised and self-supervised learning is sometimes blurred, unsupervised learning has typically been used to describe dimensionality reduction, clustering, and generative modeling algorithms. Self-supervised learning (SSL) describes a special case of unsupervised learning where the system derives *labels* from the data itself (Balestriero, Ibrahim, 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. Most famously, SSL is the backbone to the success of Large Language Models (Vaswani et al., 2017) and in particular ChatGPT (OpenAI, 2021), a language model trained on a pretext task of predicting masked words in a sentence. SSL methods have also recently been shown to outperform supervised methods for certain computer vision tasks for large datasets (Goyal et al., 2019).

2.1 Multiview Machine Learning

This thesis is focussed on multiview machine learning. Here, data from different sources or modalities, referred to as views, such as neuroimaging, genomics, and clinical records, are analyzed collectively to unveil underlying patterns. 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 broadly fall into categories of supervised and self-supervised multiview learning, with some algorithms straddling the boundary between the two.

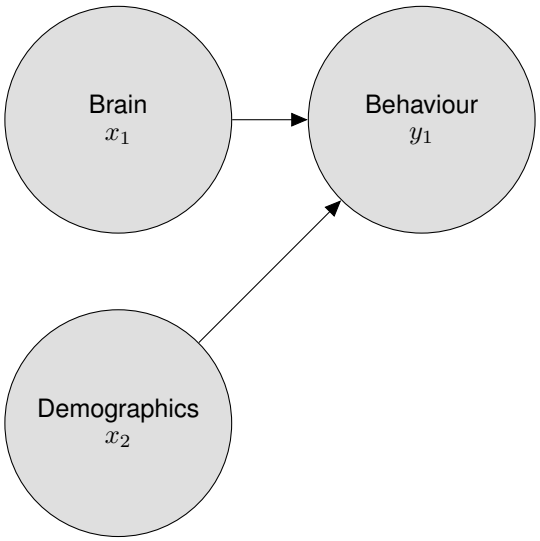


Figure II.1: ***Supervised multiview learning in mental health:*** An example of how different views (brain activity and demographics) can be fused to predict a target variable (behavior) in a supervised learning setting.

2.1.1 Supervised Multiview Learning

In supervised multiview learning, the goal is to fuse information from multiple distinct views or feature sets to improve the predictive performance of a model. This approach involves integrating the various views, often using one view as the target variable and the others as predictors. The model learns to combine the information from the different views to make more accurate predictions than would be possible using any single view alone.

For instance, in the context of mental health, we can consider behavioral data as a dependent variable influenced by multiple independent variables like brain activity and demographics. Figure II.1 illustrates this concept, where behavioral patterns (y_1) are predicted based on features from brain activity (x_1) and demographic information (x_2). The model learns to fuse the information from the brain and demographic views to form a more comprehensive understanding of the factors influencing behavior.

Multiple Kernel Learning (MKL) (Gönen and Alpaydın, 2011) is a prominent example of supervised multiview learning, where the algorithm learns to combine kernel representations of the different views. By fusing the information from the various kernels, MKL enhances the model's predictive capabilities compared to using a single kernel.

With the advent of deep learning, the underlying concept of MKL has been extended to deep learning architectures. These architectures enable the model to learn and fuse representations from various views more effectively (Guo, J. Wang, and S. Wang, 2019). The deep learning models can automatically learn the most informative features from each view and combine them in non-linear ways to make predictions.

I contributed to this line of research through the software package Fusili (Townsend, Chapman, and Cole, 2023), which implements a number of deep-learning based multi-modal data fusion methods for supervised learning. Fusili provides a flexible framework for building models that can fuse information from various data modalities, such as images, text, and structured data, to make more accurate predictions.

In summary, supervised multiview learning involves fusing information from multiple views to improve predictive performance. By combining the unique perspectives offered by each view, these methods can form a more comprehensive understanding of the problem and make more accurate predictions than would be possible using any single view alone.

2.1.2 Self-Supervised Multiview Learning

In contrast to supervised multiview learning, where explicit labels guide the learning process, self-supervised multiview learning operates under the hypothesis that different views are manifestations of a shared, yet hidden, latent variable (Zong, Mac Aodha, and T. Hospedales, 2023). This approach, as evidenced in the latent variable model of mental health illustrated in Figure II.2, suggests that both neuroimaging and behavioural data are influenced by an underlying factor, such as the severity of a mental health condition, which remains unobserved.

A key challenge in self-supervised learning is designing pretext tasks to estimate this latent source from the available views. A common approach is to estimate a common low-dimensional representation of the variance in the data from both views. In most objectives of this form, this ammounts to identifying the mutual information between the views. These representations may be informative for their own sake, identifying common factors between the views, or they may be used as inputs to a downstream task, such as classification or regression.

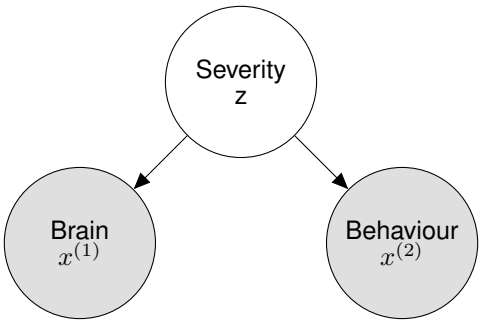


Figure II.2: ***Latent Variable Model of Mental Health:*** From this perspective the neuroimaging modality and behavioural data are both considered to have been generated with distributions conditioned on the severity of a mental health condition

2.2 Conditional Independence, Causality, and Multiview Learning

The graphical model in Figure II.1 represents the assumption that the brain and demographics are independent variables, and that the behaviour is a conditional variable, dependent on both the brain and demographics.

On the other hand, the graphical model in Figure II.2 represents the assumption that the brain and behaviour are conditionally independent given the severity of an unobserved latent mental health condition.

Reichenbach (1956) introduced the epoynmous Reichenbach's principle, which states that if two variables are correlated, then either one causes the other, or both are caused by a third variable. While the relationship between conditional independence and causality is nuanced (Pearl, 2009), it is clear that our assumptions about the causal structure of the data can inform our choice of multiview learning algorithm. In particular, we could envision a number of causal structures that could give rise to the observed data in Figure II.2:

- 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

In the first case, we might be more inclined to use a supervised multiview learning algorithm to predict one view from the other. In the second case, we might be more

inclined to use a self-supervised multiview learning algorithm to estimate the latent variable.

2.2.1 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; Lyu et al., 2021; M.-S. Chen et al., 2022). The complementary information in views offers unique insights into different aspects of the same subject. For instance, in mental health studies, neuroimaging might reveal structural changes in the brain that are not (yet) present in presented behavioural phenotypes, while behavioral data could be influenced by demographic factors that do not present as structural differences in the brain. Both these views together provide a more holistic understanding of a mental health condition. Conversely, redundant information in different views refers to overlapping or similar data presented from various angles. For instance, a specific mental health condition may manifest in both observable behavioral changes and detectable neuroimaging markers. While each view alone could suggest the presence of the condition, their combination, due to redundancy, can enhance the reliability of the diagnosis. This redundancy is not merely repetitive; it plays a crucial role in denoising and validating findings. In essence, if both neuroimaging and behavioral data independently point to the same diagnosis, the confidence in this diagnosis increases. The *Wisdom of Crowds* phenomenon, where the collective average of multiple estimates tends to be more accurate than individual estimates, exemplifies the strength of redundant information (Galton, 1907), as illustrated in Figure II.3. This principle is akin to the redundancy in multiview data, where multiple views converge to a more accurate or robust conclusion than any single view alone.

In this thesis, we will explore Canonical Correlation Analysis, a multiview learning method predicated on the assumption that different views provide complementary information about latent variables. The following sections will establish a formal framework for representation learning and motivate the use of Canonical Correlation Analysis in harnessing complementary information from multiview data.

Figure II.3: ***The Wisdom of Crowds:*** The average of multiple noisy estimates of the weight of a cow is more accurate than any individual estimate

3 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 We will consistently use the superscript (i) to denote the i-th view and not as an exponentiation operation. $d \in [D_i]$ 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.

We denote the inner product between two vectors a and b as $\langle a, b \rangle$, which is defined as:

$$\langle a, b \rangle = a^\top b = \sum_{i=1}^{n} a_i b_i \tag{II.2}$$

where a_i and b_i are the i-th elements of vectors a and b, respectively, and n is the dimension of the vectors.

The inner product is a measure of similarity between two vectors, with larger values indicating higher similarity. It is also related to the angle θ between the vectors, as shown in the following equation:

$$\langle a, b \rangle = |a||b| \cos(\theta) \tag{II.3}$$

where $|a|$ and $|b|$ are the Euclidean norms (lengths) of vectors a and b, respectively.

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 (noting that in the CCA literature they are sometimes referred to as canonical variables (Borga, 1998)), 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; Alpert and R. A. Peterson, 1972; Borga, 1998), noting that the same concept has also been referred to as structure correlations (Meredith, 1964).

We will use the notation $\Sigma_{ij} = \mathrm{Cov}(X^{(i)}, X^{(j)})$ for the population covariance matrix between the random variables associated with view i and j. This covariance matrix captures the relationships between variables from different views. Each element of this matrix, $\Sigma_{ij}(a, b)$, measures how much the a-th variable in view i and the b-th variable in view j change together, even though they belong to different views. A positive covariance indicates that the variables from different views tend to increase or decrease together, while a negative covariance indicates that they tend to move in opposite directions. These covariance matrices play a crucial role in multiview learning algorithms as they capture the inter-view relationships that the algorithms aim to leverage.

We will also use $\Sigma_{ii} = \mathrm{Cov}(X^{(i)})$ for the population covariance matrix of the random variables associated with view i with each other. This covariance matrix captures the relationships between variables within the same view. Each element of this matrix, $\Sigma_{ii}(a, b)$, measures how much the a-th and b-th variables in view i change together. A positive covariance indicates that the variables within the same view tend to increase or decrease together, while a negative covariance indicates that they tend to move in opposite directions. These within-view covariance matrices are essential for understanding the structure of the data in each view and are used in conjunction with the inter-view covariance matrices in multiview learning algorithms.

Many classical subspace learning algorithms can be formulated as Generalized Eigenvalue Problems (GEPs) constructed from covariance matrices. In the following

subsection, we introduce the concept of GEPs and discuss their properties, which will be useful for understanding the optimization problems and solutions of these algorithms.

3.1 Generalized Eigenvalue Problems 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:

$$Au = \lambda Bu \tag{II.4}$$

with $\lambda \in \mathbb{R}, u \in \mathbb{R}^D$, called (generalized) eigenvalue and (generalized) eigenvector respectively. When B is positive definite, then the GEP becomes equivalent to an eigen-decomposition of the symmetric matrix $B^{-1/2}AB^{-1/2}$ (Ghojogh, Karray, and Crowley, 2019). 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 generalized eigenvalues $1 \geq \lambda_1 \geq \lambda_2 \geq \cdots \geq 0$ are unique (Mills-Curran, 1988).

4 Classical Subspace Learning Algorithms

4.1 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.

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

$$Z_k = Xu_k, \tag{II.5}$$

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

where u_k are orthonormal basis vectors such that:

$$u_k^\top u_k = 1, \quad u_k^\top u_l = 0 \text{ for } k \neq l. \tag{II.6}$$

The primary goal of PCA is to maximize the variance of the representations Z_k, finding the directions of maximal variance in the data.

4.1.1 Optimization and Solution

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

$$u_{\text{opt}} = \underset{u}{\text{argmax}} \left(u^\top \Sigma u \right) \tag{II.7}$$

subject to:

$$u^\top u = 1$$

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

To solve this constrained optimization problem, we can use the method of Lagrange multipliers. The key idea behind Lagrange multipliers is to transform a constrained optimization problem into an unconstrained one by incorporating the constraints into the objective function. The Lagrange multiplier, denoted by λ, can be thought of as a penalty for violating the constraint. By setting the Lagrange multiplier to a large enough value, we ensure that the optimal solution satisfies the constraint.

The Lagrangian function for the PCA optimization problem is constructed by adding the constraint multiplied by the Lagrange multiplier to the original objective function:

$$f(u, \lambda) = u^\top \Sigma u + \lambda(1 - u^\top u), \tag{II.8}$$

where λ is the Lagrange multiplier.

Intuitively, the first term in the Lagrangian represents the objective function (maximizing the variance), while the second term penalizes solutions that violate the constraint (unit norm). By finding the stationary points of the Lagrangian with respect to both u and λ, we obtain the optimal solution that maximizes the variance while satisfying the unit norm constraint.

Differentiating the Lagrangian with respect to u and setting it to zero yields the

first-order conditions (which are necessary for optimality for the optimal u):

$$\Sigma u = \lambda u, \tag{II.9}$$

$$u^\top u = 1. \tag{II.10}$$

Eigenvalue Problem This transforms the problem into an eigenvalue equation for the covariance matrix Σ, 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 λ. Subsequent components are the remaining eigenvectors ordered by their corresponding eigenvalues.

4.1.2 Limitations

There are three major limitations of PCA that are relevant to this thesis.

1. **Scale Invariance**: as highlighted in the epigraph to this chapter, PCA is not scale invariant, meaning that the importance of a principal component can be disproportionately affected by the scale of the variables in the data. Variables measured at larger scales can dominate over those measured at smaller scales unless the data is normalized. This sensitivity to the scale of the data can lead to misleading directions that do not necessarily capture the most meaningful underlying structures.

2. **Sparsity and Interpretability**: Although PCA reduces dimensionality by projecting the data onto new axes, the resulting principal components are linear combinations of *all* the original features. This complex aggregation can make it difficult to interpret the components, especially when each component is influenced by many original variables. For this reason, sparse variants of PCA have been developed (Zou, Hastie, and Robert Tibshirani, 2006; Zou and Xue, 2018), which aim to find sparse linear combinations of the original features; interpretable as a subset of the original features contributing to a significant proportion of the variance in the data.

3. **Multiview Data Utilization**: PCA is primarily designed for analyzing a single dataset and does not naturally accommodate multiview data, where multiple independent sets of variables (views) describe the data. While it is possible to concatenate these views into a single dataset prior to analysis, this approach does not take advantage of the potential interactions and complementary

information across the views, which can be critical for more insightful analysis in applications such as image processing, bioinformatics, and social sciences.

Nevertheless, PCA remains a popular tool in practice (Greenacre et al., 2022) and is a useful baseline for multiview learning methods, and we will use it as a point of comparison throughout this thesis.

4.2 Partial Least Squares

Given the inherent limitations of PCA, especially in handling multiview datasets where capturing interactive and complementary information between different data sources is crucial, Partial Least Squares (PLS) emerges as a potent alternative. PLS extends the principles of PCA to analyze two correlated views simultaneously, optimizing for the shared covariance rather than variance within a single dataset. This approach makes PLS particularly valuable in multiview settings where the goal is to uncover the latent structures that explain the relationships between views.

Partial Least Squares (PLS) (Wold, 1975) 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.

PLS optimizes for the dot product between the representations of two views, a measure of similarity.

$$\langle X^{(1)}u^{(1)}, X^{(2)}u^{(2)}\rangle = |X^{(1)}u^{(1)}||X^{(2)}u^{(2)}|\cos(\theta) = u^{(1)\,T}\Sigma_{12}u^{(2)} \tag{II.11}$$

Where θ is the angle between the two representations. Much like for PCA, the representations are constrained to be linear transformations of the form:

$$Z^{(i)} = X^{(i)}u^{(i)} \tag{II.12}$$

Where $u^{(i)}$ are orthonormal basis vectors such that:

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

$$u^{(i)\,T}u^{(j)} = 0 \text{ for } i \neq j \tag{II.14}$$

4.2.1 Optimization and Solution

The constrained optimization problem for PLS can therefore be formulated as:

$$u^{(1)}_{\text{opt}} = \underset{u^{(1)}}{\operatorname{argmax}}\{u^{(1)\,T}\Sigma_{12}u^{(2)}\} \tag{II.15}$$

subject to:

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

The Lagrangian for this optimization problem once again integrates the constraints as penalties:

$$f(u^{(1)}, \lambda) = u^{(1)\,T}\Sigma_{12}u^{(2)} + \lambda_1(1 - u^{(1)\,T}u^{(1)}) + \lambda_2(1 - u^{(2)\,T}u^{(2)}) \tag{II.16}$$

Upon deriving the first order conditions, we get:

$$\Sigma_{21}u^{(1)} = \lambda_2 u^{(2)} \tag{II.17}$$
$$\Sigma_{12}u^{(2)} = \lambda_1 u^{(1)} \tag{II.18}$$
$$u^{(1)\,T}u^{(1)} = 1 \tag{II.19}$$
$$u^{(2)\,T}u^{(2)} = 1 \tag{II.20}$$

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

$$\Sigma_{21}\Sigma_{12}u^{(2)} = \lambda^2 u^{(2)} \tag{II.21}$$
$$\Sigma_{12}\Sigma_{21}u^{(1)} = \lambda^2 u^{(1)} \tag{II.22}$$

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:

$$\begin{pmatrix} 0 & \Sigma_{12} \\ \Sigma_{21} & 0 \end{pmatrix} \begin{pmatrix} u^{(1)} \\ u^{(2)} \end{pmatrix} = \lambda I \begin{pmatrix} u^{(1)} \\ u^{(2)} \end{pmatrix} \tag{II.23}$$

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 $(\lambda_1, \ldots, \lambda_K)$ representing covariances as:

$$\mathrm{PLS}_K(X^{(1)}, X^{(2)}) := (\lambda_k)_{k=1}^K \tag{II.24}$$

4.2.2 Limitations

Despite the advantages of PLS over PCA in handling multiview datasets, PLS has its own limitations that can impact its effectiveness in certain applications:

1. **Scale Invariance**: Similar to PCA, PLS is not scale invariant. This means that the model's outcomes are affected by the scale of the features, potentially leading to biased weights towards features with larger scale unless data is normalized.

2. **Sparsity and Interpretability**: PLS does not inherently produce sparse models. The components derived from PLS are linear combinations of all input features, which can make the model difficult to interpret, particularly in high-dimensional contexts such as genomics or text processing. Sparse PLS has also been an active area of research (Chun and Keleş, 2010; D. M. Witten, Robert Tibshirani, and Hastie, 2009).

4.3 Canonical Correlation Analysis

Canonical Correlation Analysis (CCA) is introduced as an advancement over PLS, designed to maximize the correlation instead of covariance between representations. This focus on correlation allows for a more nuanced understanding of the relationships between views, making CCA particularly useful in scenarios where the goal is to explore how views relate on a normalized scale.

CCA achieves this by optimizing the cosine of the angle between the representations, thus normalizing the effect of the scale of the data:

$$\cos(\theta) = \frac{\langle X^{(1)}u^{(1)}, X^{(2)}u^{(2)} \rangle}{|X^{(1)}u^{(1)}||X^{(2)}u^{(2)}|} = \frac{u^{(1)\,T}\Sigma_{12}u^{(2)}}{|X^{(1)}u^{(1)}||X^{(2)}u^{(2)}|} \tag{II.25}$$

By focusing on correlation, CCA normalizes the contributions of each variable, ensuring that the analysis is not unduly influenced by the magnitude of the data. This normalization is particularly valuable in multiview settings where the scales of the data sources may differ significantly.

4.3.1 Optimization and Solution

If we constrain the norms of the representations to be equal to 1, i.e., $|X^{(1)}u^{(1)}| = |X^{(2)}u^{(2)}| = 1$, then maximizing the cosine similarity is equivalent to maximizing the numerator $u^{(1)\,T}\Sigma_{12}u^{(2)}$. This leads to the following constrained optimization problem:

$$\begin{aligned} &u_{\text{opt}} = \underset{u}{\operatorname{argmax}}\{u^{(1)\,T}X^{(1)\,T}X^{(2)}u^{(2)}\} \\ &\text{subject to:} \\ &u^{(1)\,T}\Sigma_{11}u^{(1)} = 1 \\ &u^{(2)\,T}\Sigma_{22}u^{(2)} = 1 \end{aligned} \tag{II.26}$$

By focusing on correlation and imposing unit norm constraints on the representations, CCA normalizes the contributions of each variable, ensuring that the analysis is not unduly influenced by the magnitude of the data. This normalization is particularly valuable in multiview settings where the scales of the data sources may differ significantly.

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:

$$\Sigma_{21}u^{(1)} = \lambda^{(2)}\Sigma_{22}u^{(2)} \tag{II.27}$$

$$\Sigma_{12}u^{(2)} = \lambda^{(1)}\Sigma_{11}u^{(1)} \tag{II.28}$$

$$u^{(1)\,T}\Sigma_{11}u^{(1)} = 1 \tag{II.29}$$

$$u^{(2)\,T}\Sigma_{22}u^{(2)} = 1 \tag{II.30}$$

Eigenvalue Problems Substituting the second two conditions into the first two, we get $\lambda^{(1)} = \lambda^{(2)} = \lambda$. Finally, substituting the first two conditions into each other, we find the eigenvalue problems:

$$\Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}u^{(1)} = \lambda^2 u^{(1)} \tag{II.31}$$

$$\Sigma_{22}^{-1}\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}u^{(2)} = \lambda^2 u^{(2)} \tag{II.32}$$

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:

$$u_{\text{opt}} = \underset{u}{\operatorname{argmax}}\{u^{(1)\,T}\Sigma_{11}^{-\frac{1}{2}}\Sigma_{12}\Sigma_{22}^{-\frac{1}{2}}u^{(2)}\} \tag{II.33}$$

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) through the matrix $T = \Sigma_{11}^{-\frac{1}{2}}\Sigma_{12}\Sigma_{22}^{-\frac{1}{2}}$. This form also shows that PLS and CCA can be made equivalent by whitening the data matrices before constructing the covariance matrices. 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.27 as a matrix equation:

$$\begin{pmatrix} 0 & \Sigma_{12} \\ \Sigma_{21} & 0 \end{pmatrix} \begin{pmatrix} u^{(1)} \\ u^{(2)} \end{pmatrix} = \lambda \begin{pmatrix} \Sigma_{11} & 0 \\ 0 & \Sigma_{22} \end{pmatrix} \begin{pmatrix} u^{(1)} \\ u^{(2)} \end{pmatrix} \tag{II.34}$$

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

Canonical Correlations In the case of CCA, the generalized eigenvalues λ are generally called canonical correlations (Hotelling, 1935; Hotelling, 1992). Given the notions of uniqueness in GEPs, the weights u are not in general unique but we can write the vector of generalized eigenvalues or canonical correlations as:

$$\mathrm{CCA}_K(X^{(1)}, X^{(2)}) := (\rho_k)_{k=1}^K \tag{II.35}$$

4.3.2 Limitations

A major limitation of CCA is revealed by the forms in equations II.31 and equation II.33; CCA in general requires the inversion of covariance matrices, which is computationally expensive, potentially numerically unstable, and impossible when the number of features exceeds the number of samples such that the covariance matrices are not full rank.

4.4 Multiview CCA

Multiview CCA (MCCA) is an extension of CCA that handles more than two views simultaneously. Given I views $X^{(1)}, \ldots, X^{(i)}$, the goal of MCCA is to find a set of directions $u^{(1)}, \ldots, u^{(i)}$ that maximize the sum of pairwise correlations between the projections of the views onto these directions.

4.4.1 Optimization and Solution

The optimization problem for MCCA can be formulated as follows:

$$u_{\text{opt}} = \underset{u}{\operatorname{argmax}} \sum_{i=1}^{I} \sum_{j=1, j\neq i}^{I} u^{(i)\,T} \Sigma_{ij} u^{(j)} \tag{II.36}$$

subject to:

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

Generalized Eigenvalue Problem The MCCA optimization problem can be solved by formulating it as a generalized eigenvalue problem (GEP). The GEP for MCCA can be written in matrix form as follows:

$$\underbrace{\begin{pmatrix} 0 & \Sigma_{12} & \cdots & \Sigma_{1I} \\ \Sigma_{21} & 0 & \cdots & \Sigma_{2I} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{I1} & \Sigma_{I2} & \cdots & 0 \end{pmatrix}}_{A} \begin{pmatrix} u^{(1)} \\ u^{(2)} \\ \vdots \\ u^{(I)} \end{pmatrix} = \lambda \underbrace{\begin{pmatrix} \Sigma_{11} & 0 & \cdots & 0 \\ 0 & \Sigma_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma_{II} \end{pmatrix}}_{B} \begin{pmatrix} u^{(1)} \\ u^{(2)} \\ \vdots \\ u^{(I)} \end{pmatrix}. \quad \text{(II.37)}$$

The matrix A contains the cross-covariance matrices between the views, while the matrix B is a block diagonal matrix containing the within-view covariance matrices. The solution to the GEP gives the optimal directions $u^{(1)}, \ldots, u^{(I)}$ and the corresponding generalized eigenvalues λ.

Unified Framework The GEP formulation of MCCA can be further generalized to include ridge regularization, which helps stabilize the solution when the covariance matrices are ill-conditioned. This leads to a unified framework that encompasses both CCA and its ridge-regularized extensions.

Let $A, B_\alpha \in \mathbb{R}^{D \times D}$ be block matrices defined as follows:

$$A_{ij} = \mathrm{Cov}(X^{(i)}, X^{(j)}) \text{ for } i \neq j, \quad \text{(II.38)}$$

$$B_{\alpha,ii} = \alpha_i I_{D^{(i)}} + (1 - \alpha_i)\mathrm{Var}(X^{(i)}), \quad \text{(II.39)}$$

where $\alpha \in [0, 1]^I$ is a vector of ridge penalty parameters. Setting $\alpha_i = 0 \, \forall i$ recovers the standard CCA, while $\alpha_i = 1 \, \forall i$ yields the PLS solution.

In the case of standard CCA (i.e., $\alpha = 0$), we can define the MCCA correlation vector as:

$$\mathrm{MCCA}_K(X^{(1)}, \ldots, X^{(I)}) = (\lambda_1, \ldots, \lambda_K), \quad \text{(II.40)}$$

where $\lambda_1, \ldots, \lambda_K$ are the top-K generalized eigenvalues. These eigenvalues represent the average of the top-K correlations between each pair of views.

4.5 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(Balakrishnama and Ganapathiraju, 1998; Riffenburgh, 1957), 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.

4.5.1 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.

4.6 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 use 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 $\mathbf{X}^{(1)} \in \mathbb{R}^{D_1 \times b}$ and $\mathbf{X}^{(2)} \in \mathbb{R}^{D_2 \times b}$ 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. For the ease of both reader and writer, we will drop the bars for the remainder of the thesis and assume that all data matrices are centered without loss of generality.

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 (chunks of data) in later chapters 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.

5 Practical Frameworks for Evaluating Multiview Learning Methods

At this point, we have introduced the theoretical foundations of multiview learning, and a number of classical representation learning algorithms including CCA and its variants. However, it is not yet clear how we should evaluate these methods in practice. In this section, we compare the machine learning and the statistical

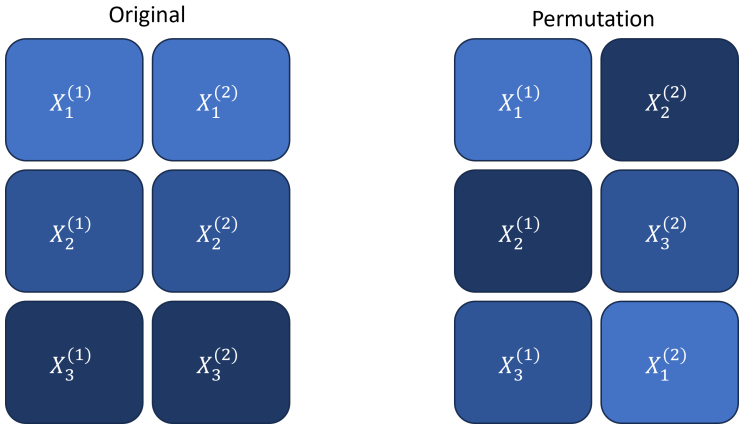


Figure II.4: Schematic of the permutation testing procedure. The original data are randomly shuffled, and the model is retrained on the shuffled data. This process is repeated multiple times, and the model's performance on the original data is compared to the distribution of performances on the shuffled data.

approach of permutation testing. These two approaches are not mutually exclusive, and statistical learning theory has emerged as a unifying framework for both perspectives (Vapnik, 1999; Hastie et al., 2009).

5.1 Permutation Testing

Permutation testing offers a robust way to evaluate the significance of the results obtained by multiview learning methods and, for a single component, is a relatively simple process (Winkler et al., 2020). As illustrated in Figure II.4, the views are randomly and separately shuffled, and the model is then trained and tested on this permuted data. This process is repeated multiple times, generating a distribution of performance metrics under the null hypothesis, where there is no relationship between the views. The actual performance of the model on the unshuffled data is then compared to this distribution. If the actual performance is significantly better than the permuted performance, it suggests that the model is capturing meaningful relationships in the data.

5.2 Machine Learning

The machine learning approach to evaluating multiview learning methods is to use a holdout or test set to estimate the out-of-sample performance of the model.

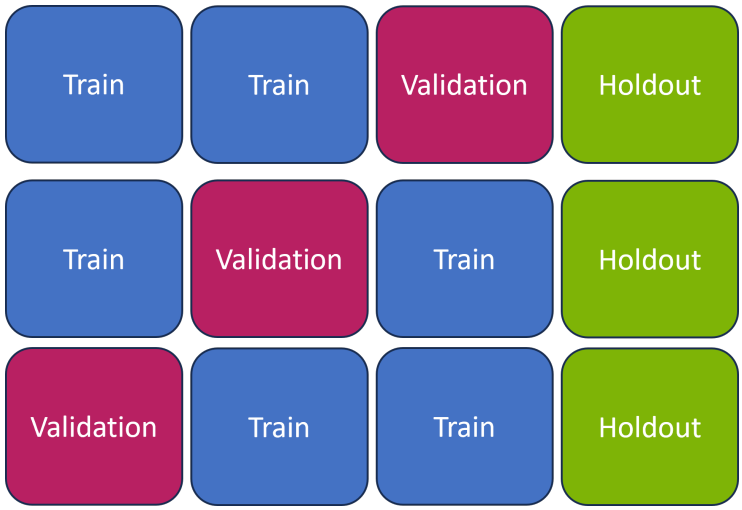


Figure II.5: Schematic of the cross-validation procedure. The original data are partitioned into training and test sets. In cross-validation, the training set is further partitioned into training and validation sets. The model is trained on the training set and evaluated on the validation set for different parameter values. The parameter value with the best performance on the validation set is selected, and the model is retrained on the entire training set. The final model is evaluated on the single test or holdout set.

Within the training set, where necessary, cross-validation is used to select the best model hyperparameters. Cross-validation involves partitioning the training set into training and validation sets, training the model on the training set, and evaluating the model on the validation set. When this is performed for multiple subsets of the training set, it is referred to as k-fold cross-validation as illustrated in figure II.5. The model hyperparameters are then selected based on the performance across the validation sets. The model is then retrained on the entire training set using the best hyperparameters, and evaluated on the test set.

In this thesis, we will use the machine learning approach throughout. This is because in scaling up to large datasets, permutation testing becomes computationally intractable. This is because permutation testing requires retraining the model multiple times on the permuted data. This comes at the cost of only being able to evaluate models with a point estimate of performance, rather than a distribution.

5.3 Components and Subspaces in CCA

5.3.1 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.

5.3.2 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.

5.3.3 Non-Uniqueness of Components

Furthermore, non-uniqueness is a significant challenge in representation learning, 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 (introduced in chapter IV), 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 is 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.

6 Multiview Learning in Neuroimaging

Finally, we review important applications of multiview learning from the literature in neuroimaging, which will be our reference in chapters III and IV.

[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.

6.1 Multiview Data in Neuroscience and Genetics

In neuroscience and genetics, two specific types of multiview studies are particularly relevant to this thesis: brain-behavior studies and imaging-genetics. Both involve the integration of data from multiple sources, offering rich insights into complex phenomena.

Brain-behavior studies typically involve pairing neuroimaging data, such as that obtained from Structural MRI (sMRI) or Functional MRI (fMRI), with non-imaging data like responses from questionnaires, cognitive test results, and other behavioral assessments. sMRI provides detailed anatomical brain images, essential for understanding brain structure and neurological disorders (Kanai and Rees, 2011), while fMRI focuses on brain function by mapping activity during cognitive tasks (Miranda et al., 2021). The integration of these imaging techniques with behavioral data offers a comprehensive view of how brain structures and functions correlate with behavioral and cognitive patterns (Rypma and D'Esposito, 2001; Genon, Eickhoff, and Kharabian, 2022).

Imaging-Genetics, another critical multiview approach, combines neuroimaging data with genetics and omics information (Lê Cao et al., 2008). This interdisciplinary field seeks to understand the genetic influences on brain structure and function, thereby illuminating the genetic basis of neuropsychiatric disorders and cognitive traits (R. Bogdan et al., 2017). Studies in this area can explore how specific genetic variations correlate with differences in brain morphology or activity patterns observed in neuroimaging (J. Liu and Calhoun, 2014).

Together, these multiview approaches are fundamental in advancing our understanding of the brain's structure, function, and its interactions with genetic and behavioral factors. They represent key applications of SSL in neuroscience and genetics, providing comprehensive insights that underpin developments in these fields.

6.2 Applications of 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. Owing to the high dimensionality of neuroimaging data, regularisation has been a particular focus of multiview learning in neuroimaging. Mihalik, Chapman, Rick A Adams, et al. (2022a) reviews the application of CCA to neuroimaging data and highlights the importance of regularisation in this context. Bilenko and Gallant (2016) 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).

7 Conclusion

In this chapter, we have provided a comprehensive overview of multiview learning, with a particular focus on its applications in neuroimaging and genetics. We have discussed the fundamental concepts and methods in multiview learning, such as Canonical Correlation Analysis (CCA), Partial Least Squares (PLS), and their variants, highlighting their strengths and limitations.

The review has emphasized the importance of regularization techniques in high-dimensional settings, as well as the challenges associated with interpreting the resulting components. We have also touched upon the need for efficient algorithms that can handle large-scale datasets, and the potential of non-linear extensions of CCA and joint embedding self-supervised learning approaches.

Furthermore, we have discussed the practical frameworks for evaluating multi-view learning methods, comparing the traditional statistical approach of permutation testing with the machine learning approach of cross-validation and holdout testing. We have also considered the complexities of identifying and interpreting top-k subspaces in CCA, and the reasons for focusing on the top-1 component in this thesis.

The applications of multiview learning in neuroimaging and genetics have been highlighted, with a particular emphasis on brain-behavior studies and imaging-

genetics. These studies have demonstrated the potential of multiview learning in uncovering the complex relationships between brain structure, function, genetics, and behavior, thereby advancing our understanding of neurological disorders and cognitive traits.

Despite the significant progress made in multiview learning, several challenges remain. These include the need for more interpretable and regularized methods, particularly in high-dimensional settings, the development of efficient algorithms for handling large-scale datasets, and the extension of CCA to non-linear and deep learning-based approaches.

In the following chapters, this thesis aims to address these challenges by proposing novel methods and techniques for multiview learning. We will explore regularized and interpretable extensions of CCA, develop efficient algorithms for high-dimensional data, and investigate the potential of deep learning-based approaches for multiview learning. By tackling these challenges, we hope to contribute to the advancement of multiview learning and its applications in neuroimaging, genetics, and beyond.

Chapter III

Regularisation of CCA Models: A Flexible Framework based on Alternating Least Squares

Contents

Preface

In this chapter, I build upon work presented at the OHBM conference (James Chapman, 2023) and the insights gained from a tutorial paper I co-authored, which included a series of simulations (Mihalik, Chapman, Rick A Adams, et al., 2022a).

1 Introduction

This chapter introduces a novel approach for analyzing large-scale neuroimaging datasets, such as the Human Connectome Project (HCP (Van Essen et al., 2013)) and Alzheimer's Disease Neuroimaging Initiative (ADNI), to understand the relationship between brain structure, function, and behavior (Stephen M. Smith and Thomas E. Nichols, 2018; Bzdok and B.T. Thomas Yeo, 2017; H.-T. Wang et al., 2020). These datasets are characterized by a disproportion between the number of subjects and the volume of features, posing a challenge for Canonical Correlation Analysis (CCA) models due to the risk of overfitting and spurious correlations (H.-T. Wang et al., 2018). For example, the HCP dataset used in this chapter contains 1003 subjects and 300 features in the functional MRI (fMRI) view while the ADNI dataset contains 592 subjects and 168,130 features in the structural MRI (sMRI) view alone.

In response to the reproducibility crisis in neuroscience (Button et al., 2013), this chapter focuses on enhancing the generalizability of CCA models through regularization, a technique that introduces a bias towards more interpretable and generalizable models (Engl, Hanke, and Neubauer, 1996; Bzdok, Thomas E Nichols, and Stephen M Smith, 2019). Existing regularization methods in CCA, such as 'sparse CCA' with Partial Least Squares (PLS) objectives (Lê Cao et al., 2008; D. M. Witten, Robert Tibshirani, and Hastie, 2009; Lindenbaum et al., 2021), are limited by their inherent bias towards the largest principal components (Mihalik, Chapman, Rick A. Adams, et al., 2022b).

To overcome these limitations, we propose the Flexible Regularised Alternating Least Squares (FRALS) framework for CCA based on the Alternating Least Squares form of CCA (Golub and Zha, 1995). FRALS allows for the integration of various regularized least squares solvers, particularly emphasizing the elastic net penalty, which combines L2 and L1 penalties. This method controls bias and promotes sparsity in model weights, advancing beyond previous sparse Brain-Behavior analysis methods.

Our application of the FRALS framework with Elastic Net regularization to the HCP and ADNI datasets showcases its effectiveness in enhancing out-of-sample canonical correlation compared to traditional CCA models. Additionally, FRALS uncovers new modes of variation in brain-behavior relationships.

In essence, this chapter presents FRALS as a robust, innovative solution for the analysis of high-dimensional neuroimaging datasets, significantly improving the reliability and interpretability of Brain-Behavior correlations.

2 Background: Regularisation for High-Dimensional and Structured Data

In this section, we review a number of regularisation 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 (Curth, Jeffares, and Schaar, 2023; Hastie et al., 2009). 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)[1]. Regularisation can be understood as a method for reducing the variance of a model by introducing a bias towards simpler models. This means regularisation can improve the generalizability of models in high-dimensional settings.

2.1.1 Implicit and Explicit Regularisation

We can implement regularisation in two different ways. Explicit regularisation is achieved by adding a penalty term to the objective function. This weights the objective function against a term that penalises complexity.

[1] 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.

Implicit regularisation is achieved by changing the optimisation algorithm and can include dimensionality reduction, as well as certain optimisation procedures like using stochastic gradient descent in place of gradient descent (Ali, Dobriban, and Ryan Tibshirani, 2020), and early stopping of optimization routines (Yao, Rosasco, and Caponnetto, 2007)

2.2 Shrinkage Regularisation

Shrinkage regularisation is a form of regularisation that penalises the magnitude of the model parameters. This technique is particularly effective in enhancing the performance of linear models in situations characterised 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.

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 be 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 generalisation.

The premise of shrinkage regularisation 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 regularisation 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.

2.2.1 PLS as Shrinkage Regularisation

PLS can be interpreted as a form of shrinkage regularisation applied to CCA. We can explain this by considering an analogy between CCA and Linear Regression[2].

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 regularisation parameter between 0 and 1[3]. The ridge penalty acts in three important ways:

- 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, Rick A Adams, et al., 2022a) (formed by reparameterizing $u^{(i)} = (\Sigma_{ii})^{-\frac{1}{2}}u^{(i)}$):

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

[3] 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

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

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:

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

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 Σ_{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 regularisation 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 regularised 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 regularisation even in these very high-dimensional settings and that more nuanced regularisation methods can offer significant improvements in performance and interpretability. PLS is evidently not a nuanced tool for regularisation because it offers no control over the degree of regularisation applied.

2.2.2 Ridge Regularisation

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

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

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 τ_2 are the ridge regularisation parameters for the first and second views respectively. By tuning these parameters, we can control the degree of regularisation 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. Ridge CCA has been shown to be effective for neuroimaging data for both CCA (A. Tenenhaus and M. Tenenhaus, 2011; Tuzhilina, Tozzi, and Hastie, 2023; Hardoon, Szedmak, and Shawe-Taylor, 2004) and Kernel CCA (Hardoon, Mourao-Miranda, et al., 2007).

2.2.3 PCA-CCA

PCA be can be used as an implicit regularisation 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. While PCA and Independent Component Analysis (ICA) are often used as preprocessing steps for CCA, they can also be used as regularisation methods in their own right. Of particular note in neuroimaging are studies with a data-driven approach to the PCA step, where the number of principal components is chosen based on the data (Z. Liu et al., 2022; Mihalik, Chapman, Rick A. Adams, et al., 2022b).

2.2.4 A Visual Comparison of Shrinkage Techniques

The distinct effects of Ridge and PCA on the eigenvalues of the effective covariance matrices can be clearly visualised with a simple visualisation. We plot the eigenvalues of covariance matrices as perceived by models with different regularisation

techniques [4]. As shown in Figure III.1, Ridge regularisation 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).

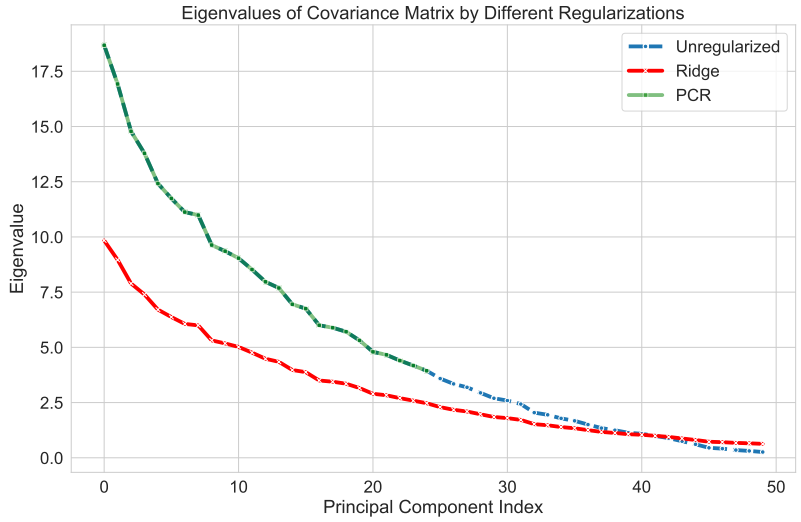


Figure III.1: Comparison of the effect of OLS, Ridge, and PCA regularisation on the eigenvalues of the covariance matrix.

When these effective covariance matrices are inverted to form the CCA objective, these effects are reversed. Ridge regularisation 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 visualisation underscores the intrinsic nature of each regularisation method:

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

[4] 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

- **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 Regularisation

Sparse regularisation is a powerful tool for improving the performance and interpretability of linear models. Sparse regularisation 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 regularisation 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[5]. 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 regularisation, we review a number of approaches to adding sparsity to the CCA problem.

2.3.1 Sparse PLS: Penalised Matrix Decomposition

Penalised Matrix Decomposition (PMD) (D. M. Witten, Robert 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 optimisation problem for PMD is then given by:

[5]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 [6]. Like the relationship between PLS and CCA, PMD and a form of CCA with constrained L1 norm are equivalent only when the covariance matrices are identity matrices. 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 optimisation, another class of approaches have adopted a penalised least squares approach.

2.3.2 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 maximised when the squared distance between $X^{(1)}u^{(1)}$ and $X^{(2)}u^{(2)}$ is minimised. (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 X. Zhang, 2019 separately proposed iterative penalised least squares methods for sparse CCA.

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

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

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 penalises the norm of the vector u. (Mai and X. Zhang, 2019) proved that solving 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 regularisation but the subproblems no longer correctly optimize the global objective[7].

2.3.3 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 regularised 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^{th} view, and T denotes the latent variable matrix. The premise is that when T closely mirrors $X^{(i)}U^{(i)}$ across

[7] when rescaling the penalised solutions back to unit variance

all i, the scores correlate. Notably, this method is adaptable to multiple views. The authors employed proximal gradient descent for regularisation, specifically suited for penalties like the lasso. While these methods are flexible, they don't have the plug-and-play nature of the penalised least squares methods. Not just a matter of convenience, this means that these methods are not compatible with existing solvers for regularised least squares problems like for example total variation regularisation solvers in nilearn, which are often highly optimised for specific problems and modalities.

2.3.4 Structured Regularisation

As highly structured data, linear models using both structural MRI and fMRI data have been shown to benefit from structured regularisation methods but notably these methods have not been applied to CCA. Total variation regularisation, 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) regularisation, 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; Cuingnet et al., 2012).

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 regularisation method to their data and research question, enhancing the interpretability and performance of Brain-Behaviour association models. This led us to propose the Flexible Regularised Alternating Least Squares (FRALS).

3 Methods: Flexible Regularised Alternating Least Squares (FRALS)

The primary goal of our Flexible Regularised 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 regularised 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 democratises 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 optimised 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[8]. 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 regularisation $\lambda_i P_i$ on the weights $u^{(i)}$:

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

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

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

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 $u^{(i)}$:

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

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

are regularised least squares problems that can be solved using any suitable regularised least squares solver[9].

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 regularised least squares solvers to solve regularised CCA problems.

4 Experiment Design

This section outlines the methodologies used in our study to explore the Flexible Regularized Alternating Least Squares (FRALS) and associated techniques in Canonical Correlation Analysis (CCA). We focus on fitting a single latent dimension for the analyses.

4.1 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, Rick A Adams, et al., 2022a). We are particularly interested in the performance of an Elastic Net FRALS on these datasets as Ridge CCA has been shown to outperform PLS (Mihalik, Chapman, Rick A Adams, et al., 2022a), implying that shrinkage regularisation is beneficial, and Sparse PLS has been shown to outperform PLS (João M Monteiro et al., 2016), implying that sparsity is beneficial. We therefore expect that Elastic Net FRALS will outperform PLS, Ridge CCA, and Sparse PLS on these datasets.

[9] 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

4.2 The Predictive Framework for CCA

Our evaluation of CCA models used a standard predictive framework, dividing the data into an 80:20 ratio for training and testing. This method ensures fitting the model on the training set without incorporating information from the test set.

4.2.1 Model Comparisons

The experiment aims to demonstrate the effectiveness of tunable shrinkage and sparsity in CCA models, enabled by the FRALS framework. We compare the performance of Elastic Net FRALS with other CCA variants such as PCA, PLS, Ridge CCA, SPLS, and Elastic Net CCA, particularly in the context of high-dimensional datasets like HCP and ADNI.

Table 4.1: Employed CCA Variants

Model	Abbreviation	Hyperparameters	Hyperparameter Range
Principal Component Analysis	PCA	-	-
Regularised 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	τ_1, τ_2	0-1 (log scaled)

4.2.2 Model Selection

For models that require hyperparameter tuning, a grid search was employed to find the best hyperparameters. We used 5-fold cross-validation to assess the performance of each model with various hyperparameters across different training data splits. The optimization goal was to achieve the highest average out-of-sample correlation.

4.2.3 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 release of the HCP. The rs-fMRI data provided brain connectivity matrices. These were derived from pairwise partial correlations between subject components obtained through group independent component analysis (ICA), utilizing 25 components. This resulted in 300 brain variables, corresponding to the lower triangle

Table 4.2: HCP Data Parameters

Parameter	Value
Number of samples (n)	1003
Number of features in View 1 (p)	300
Number of features in View 2 (q)	145

of the connectivity matrix. In our analysis, 145 non-imaging subject measures were incorporated, similar to prior studies, with the exception of 13 measures (ASR_Aggr_Pct, ASR_Attn_Pct, ASR_Intr_Pct, ASR_Rule_Pct, ASR_Soma_Pct, ASR_Thot_Pct, ASR_Witd_Pct, DSM_Adh_Pct, DSM_Antis_Pct, DSM_Anxi_Pct, DSM_Avoid_Pct, DSM_Depr_Pct, DSM_Somp_Pct) that were unavailable in the 1200-subject data release. Furthermore, nine confounding variables, including the acquisition reconstruction software version, a summary statistic of head motion during rs-fMRI acquisition, weight, height, systolic and diastolic blood pressure, hemoglobin A1C level, and cube-root of total brain and intracranial volumes as estimated by FreeSurfer, were regressed out from both data types. More details can be found in Stephen M Smith et al. (2015) and Mihalik, Chapman, Rick A Adams, et al. (2022a). We summarize the parameters of the HCP data in table 4.2.

4.2.4 The Alzheimer's Disease Neuroimaging Initiative (ADNI)

Accessible at `adni.loni.usc.edu`, the ADNI database was initiated in 2003. Its primary aim is the examination of how well serial MRI, PET (Positron Emission Tomography), biological markers, along with clinical and neuropsychological assessments, track the progression of Mild Cognitive Impairment (MCI) and the early stages of Alzheimer's disease. In our study, we utilised data from a subset of 592 unique individuals, comprising 309 males (average age 74.68 ± 7.36 SEM) and 283 females (average age 72.18 ± 7.50 SEM). This subset included 147 healthy controls, 335 individuals with Mild Cognitive Impairment (MCI), and 110 diagnosed with dementia. T1 weighted structural MRI (sMRI) scans were the source of whole-brain voxel-based grey matter volumes. The sMRI data underwent preprocessing with SPM12 (Ashburner et al., 2014), which involved segmentation, normalisation using DARTEL, reslicing to a resolution of $2 \times 2 \times 2\,\text{mm}^3$, and spatial smoothing using a Gaussian kernel with 2 mm full width at half maximum (FWHM). A grey matter voxel selection mask, with a threshold of $\geq$10%, was applied to all participants' scans, resulting in 168,130 brain variables. The Mini-Mental State Examination (MMSE) is a widely recognised neurocognitive test comprising 30 questions across

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

five cognitive domains(M. F. Folstein, S. E. Folstein, and McHugh, 1975): orientation (questions 1-10), registration (questions 11-13), attention and calculation (questions 14-18), recall (questions 19-21), and language (questions 22-30). An additional item was included in our study to account for the number of attempts a subject needed to correctly respond to the registration domain questions, leading to a total of 31 variables. As in João M Monteiro et al. (2016), no confounds were removed from these data. We summarize the parameters of the ADNI data in table 4.3.

4.3 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 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.

4.3.1 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. Since the HCP and ADNI data are high-dimensional, we drop CCA from the analysis since it would produce random results.

4.3.2 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

Table 4.4: Employed CCA Variants

Model	Abbreviation	Hyperparameters	Hyperparameter Range
Principal Component Analysis	PCA	-	-
Regularised 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	τ_1, τ_2	0-1^{10} (log scaled)

splits of the training data with non-overlapping validation sets. We optimise for the hyperparameters that give the best average out of sample correlation.

5 Experiment Results

5.1 HCP Results

Next, we consider the results of applying the various CCA variants to the HCP data.

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.2). This suggests that tunable L2 regularisation 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.

Nonetheless, the Elastic Net model demonstrated a more sparse representation than the Ridge CCA model, with the Elastic Net model utilizing 241 and 96 non-zero weights for the brain and behaviour views, respectively (Table 5.1). In contrast, the Ridge CCA model used 300 and 145 non-zero weights for the respective views. Moreover, the SPLS model achieved an even sparser solution with only 118 and 56 non-zero weights for the brain and behaviour views. Considering the comparable performance of the Elastic Net and Ridge CCA models, the former's sparsity may offer a preferable solution.

5.1.2 Behaviour Weights

Figure III.3 presents the top eight positive and negative non-imaging weights for each model to visualize the behavioural data variations observed in the previous

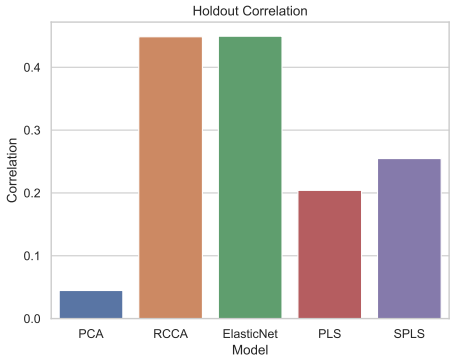


Figure III.2: HCP: Comparative out-of-sample canonical correlations among PCA, RCCA, ElasticNet, PLS, and SPLS models. The bars represent the correlation coefficients, indicating that Ridge CCA and Elastic Net models have superior performance over PLS and SPLS in capturing holdout correlation.

Table 5.1: HCP: Sparsity of models reflected by the count of non-zero weights. Elastic Net and SPLS demonstrate increased sparsity in the model weights for both brain and behaviour views, compared to PCA, RCCA, and PLS.

Model	Brain Weights (out of 300)	Behaviour Weights (out of 145)
PCA	300	145
RCCA	300	145
Elastic Net	241	96
PLS	300	145
SPLS	118	56

section. The PCA model emphasizes a mode of variation with positive correlations to psychiatric and life function tests, contrasting with negative correlations to certain emotion and personality tests. In comparison, the RCCA and Elastic Net models highlight a variation mode negatively correlated with the Line Orientation test and to a lesser extent, smoking, while showing positive correlations with other cognitive assessments. The PLS model's variation mode echoes the positive-negative pattern identified by Stephen M Smith et al., 2015, showing positive correlations with agreeableness, vocabulary tests, and life satisfaction, juxtaposed with strong negative correlations with smoking and antisocial behaviors. SPLS selects a similar mode but prioritizes vocabulary tests and smoking over rule-breaking and antisocial personality traits, aligning with the preprocessing steps described in Stephen M Smith et al., 2015 which incorporated a top-100 PCA projection of both brain and

behavioural 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. This was achieved by assigning each node to the network with the highest voxelwise overlap. These are then arranged around the circumference of the chord diagram using the Nichord package (P. C. 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. The text color for each network matches the color of the corresponding region on the outside of the chord diagram, providing a helpful visual guide.

- The **RCCA** model displays a diverse set of connections across all networks, with especially prominent weights in the somatomotor and default mode networks.
- The **ElasticNet** model presents similar connections between the somatomotor and default mode networks.
- The **PLS** model exhibits strong connections between the frontoparietal and visual networks.
- The **SPLS** model exhibits similar connections between the frontoparietal and visual networks.

This is perhaps consistent with the behaviour data as the somatomotor network is associated with motor function and sensory processing which is related to the Line Orientation test, requiring spatial reasoning and motor coordination.

The correlations made by the PLS and SPLS models between substance abuse and cognitive tests could be due to the significant role the frontoparietal network plays in executive function, which can be impaired by substance abuse. Likewise,

the visual network is likely involved in a number of the cognitive tests and could be disrupted by substance abuse.

The RCCA and ElasticNet models might be detecting more integrative and possibly higher cognitive functions, while the PLS and SPLS models might be highlighting the more immediate cognitive processes that can be disrupted by substance abuse.

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 III.5, 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.

5.2 ADNI Results

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

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

Model	Brain Weights (out of 168130)	Behaviour Weights (out of 31)
PCA	168130	31
RCCA	168130	31
Elastic Net	59617	17
PLS	168130	31
SPLS	74995	10

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 emphasize 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 A.2. Previous work using SPLS with the ADNI dataset identified the same striking pattern of weights with the model strikingly selecting the hippocampal weights (João 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.

6 Discussion and Limitations

The Flexible Regularised Alternating Least Squares (FRALS) framework for CCA, introduced in this chapter, exhibits promising performance in terms of out-of-sample correlation. Our findings indicate that, while Elastic Net CCA generally outperforms other CCA variants, much of the benefit is derived from using properly tuned Ridge regularization. This is most obviously illustrated in the HCP dataset where sparsity does not appear to be beneficial in terms of out-of-sample correlation and therefore casts doubt on whether the sparsity of the model is interpretable or spurious. It also questions whether the additional computational cost of Elastic Net CCA is justified. Our experiments reveal that Ridge CCA typically outperforms PLS across both datasets. This observation is akin to the dynamics of regularized regression, where maximal ridge regularization is seldom necessary, even in high-dimensional contexts.

6.1 FRALS Limitations

The Flexible Regularised Alternating Least Squares (FRALS) framework, while effective in certain aspects, is notably limited by its computational inefficiency. This inefficiency arises from two main factors: the dynamic nature of regression targets and the intensive computation required for each iteration.

6.1.1 Changing Regression Targets

In FRALS, regression targets are not static but dynamically evolve during the algorithm's execution. These targets are essentially projections of the other view,

and as they change, they alter the optimization landscape. Consequently, the algorithm must frequently recompute the least squares solution for each view. This process results in significant computational overhead and often leads to redundant calculations, thereby contributing to the inefficiency of the FRALS framework.

6.1.2 Computational Time

The primary computational challenge in FRALS is the repeated calculation of the least squares solution for each view in every iteration. This requirement is resource-intensive and is the main factor contributing to the slow speed of the FRALS algorithm. Empirical observations from our experiments show that FRALS operates at a pace approximately 10 times slower than Ridge CCA, varying with the specifics of the experimental setup. This disparity in speed is particularly noteworthy given the popularity of SPLS due to its speed and convenience. Figure III.12 provides an estimate of the time taken to fit each model across complete training datasets over ten runs. It is evident from the figure that Elastic Net CCA, despite being an iterative algorithm, is significantly slower than other models, particularly with the high-dimensional ADNI data. While SPLS demonstrates much faster processing, it is only marginally slower than PLS and RCCA, both of which employ optimized solvers in C and use PCA preprocessing for efficiency. Consequently, PCA emerges as the fastest model in these comparisons.

6.2 Conclusion

In this chapter, we introduced the Flexible Regularised Alternating Least Squares (FRALS) framework for CCA. We used the FRALS framework to implement Elastic Net CCA. We then compared the performance of Elastic Net CCA with other CCA variants on two datasets: the HCP and ADNI. We found that Elastic Net CCA outperformed other CCA variants on both datasets but that the performance of Elastic Net CCA was similar to Ridge CCA on the HCP dataset. However, we found that Elastic Net CCA was much slower than other CCA variants.

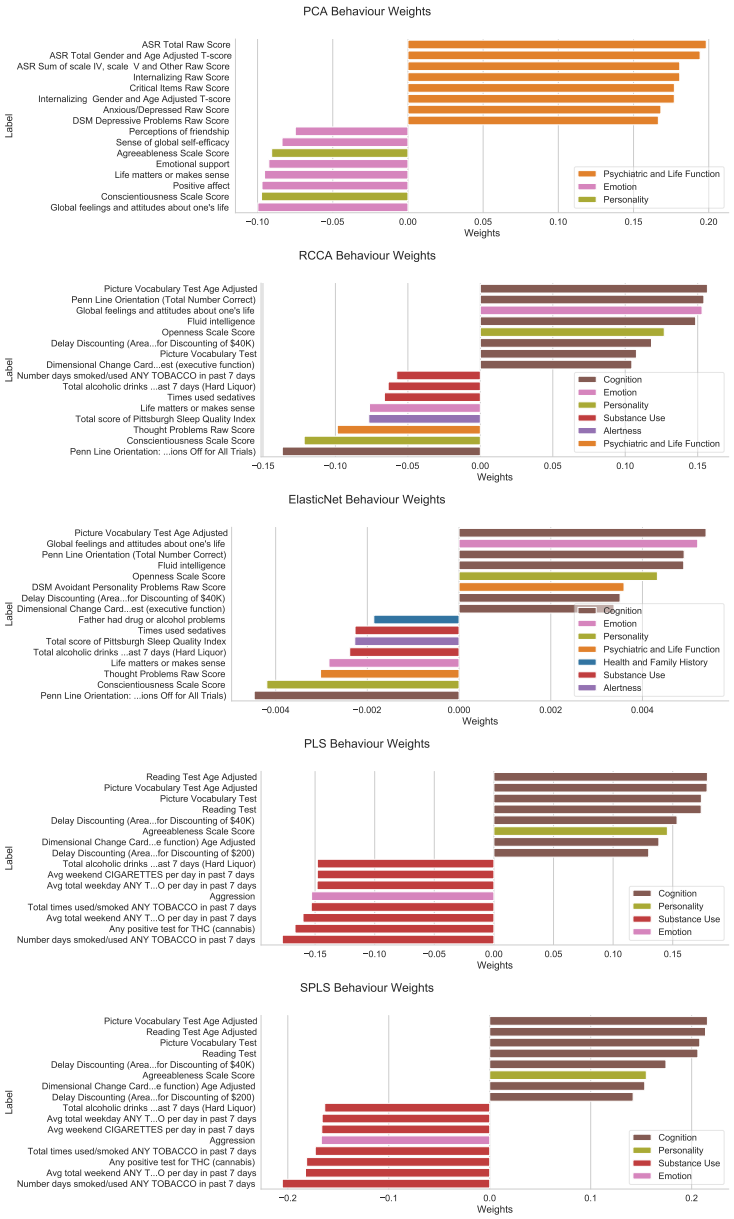


Figure III.3: HCP: Behavioural weights highlighting the top-8 positive and negative non-imaging weights. Each subfigure represents a distinct model's weight distribution across various behavioural domains such as cognition, emotion, personality, substance use, alertness, and psychiatric and life function. The variations in the weight profiles across models reflect differing patterns of association with the behavioural traits considered in the study.

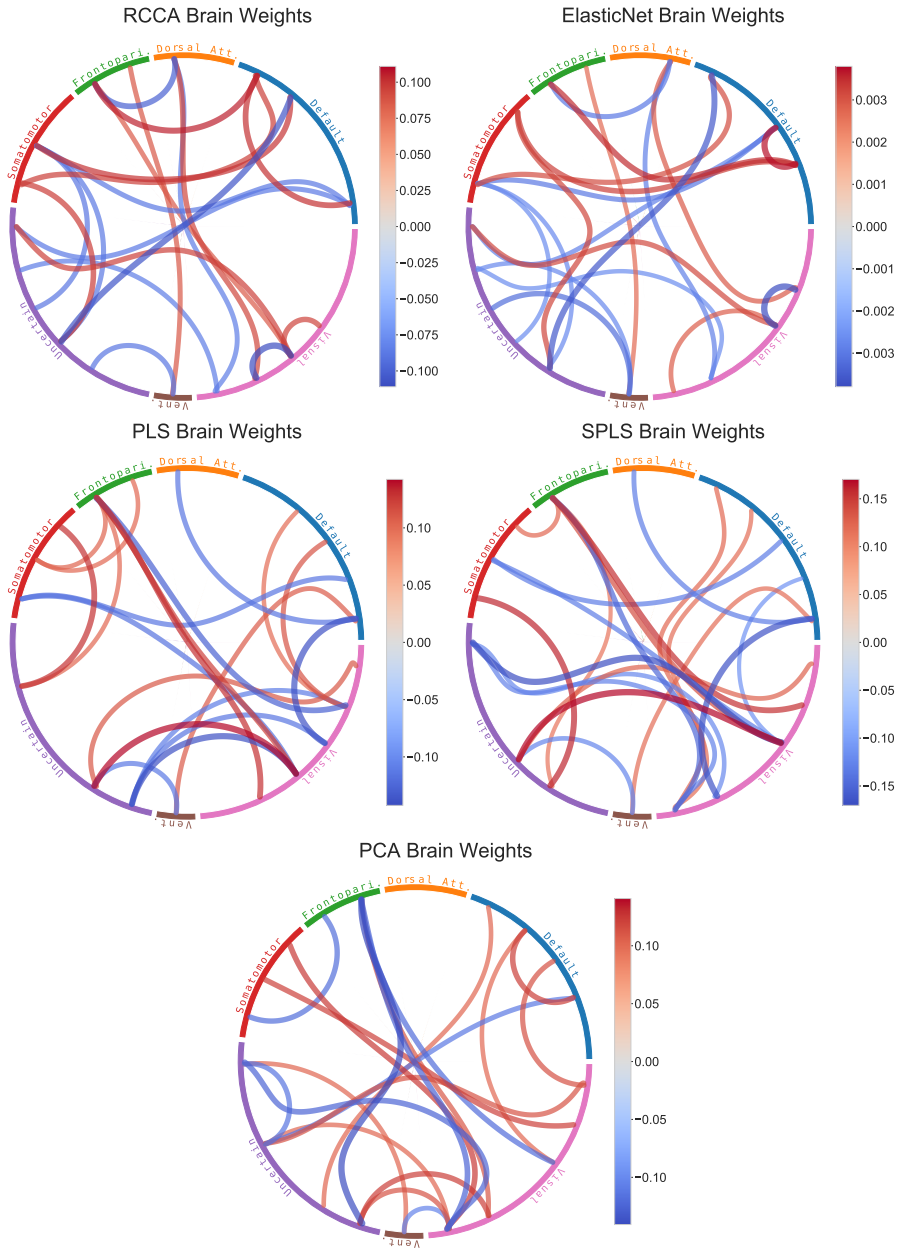


Figure III.4: HCP: Brain connectivity weights visualized through chord diagrams for multiple models. Each diagram portrays the 8 strongest positive (red to blue gradient) and negative (blue to red gradient) weights, grouped by the Yeo 7 network parcellation.

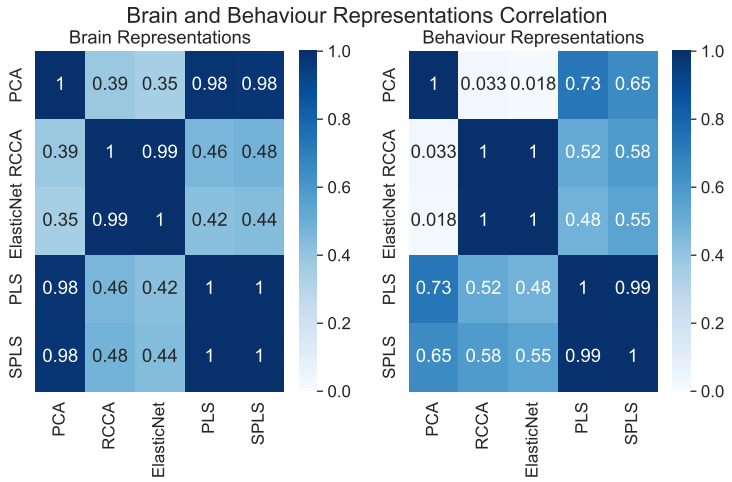


Figure III.5: HCP: Pairwise correlation matrix of brain representations across different models. The high correlation coefficients between PCA, PLS, and SPLS indicate a significant overlap in the brain representations they produce, suggesting a bias of PLS toward principal components. Contrarily, the Ridge CCA and Elastic Net models show notably lower correlations with PCA, indicating that these models capture brain representations beyond the first principal components.

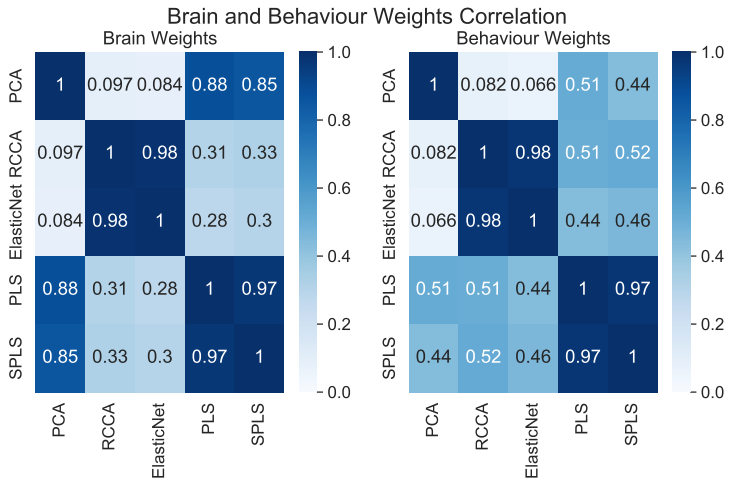


Figure III.6: HCP: Pairwise correlation matrix of the brain and behaviour weights used by each model. Similar to the brain representations, PCA, PLS, and SPLS show a high correlation in their weights, indicating similarity in the factors they consider significant. The lower correlations observed for Ridge CCA and Elastic Net with PCA suggest that these models give importance to different aspects of the data, potentially capturing more nuanced relationships.

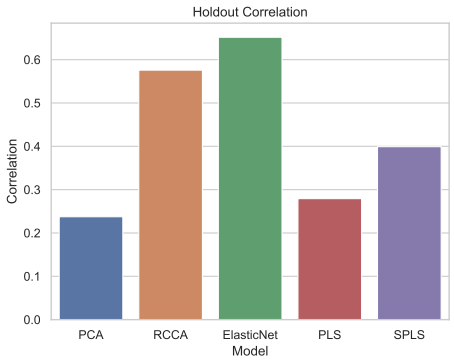


Figure III.7: ADNI: Comparative out-of-sample canonical correlations among PCA, RCCA, ElasticNet, PLS, and SPLS models. The bars represent the correlation coefficients, indicating that the Elastic Net models has superior performance over Ridge CCA, PLS, and SPLS in capturing holdout correlation.

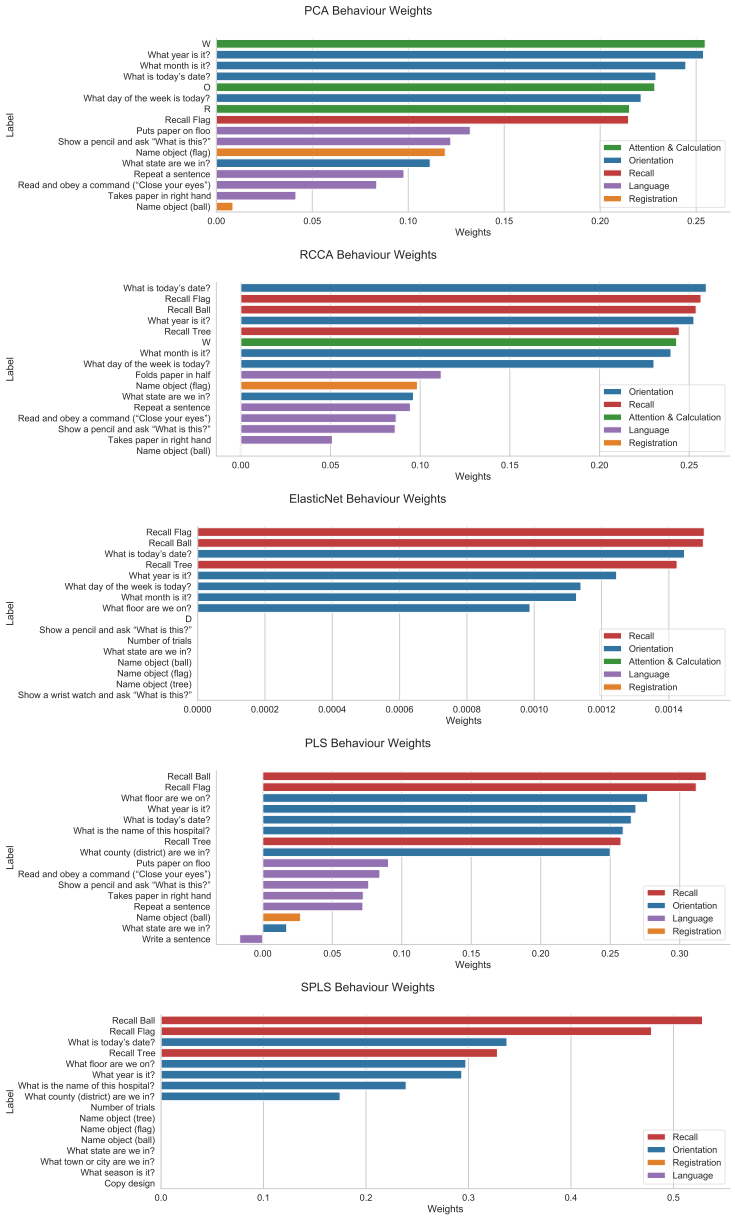


Figure III.8: ADNI: Bar plots of the behaviour weights for each model.

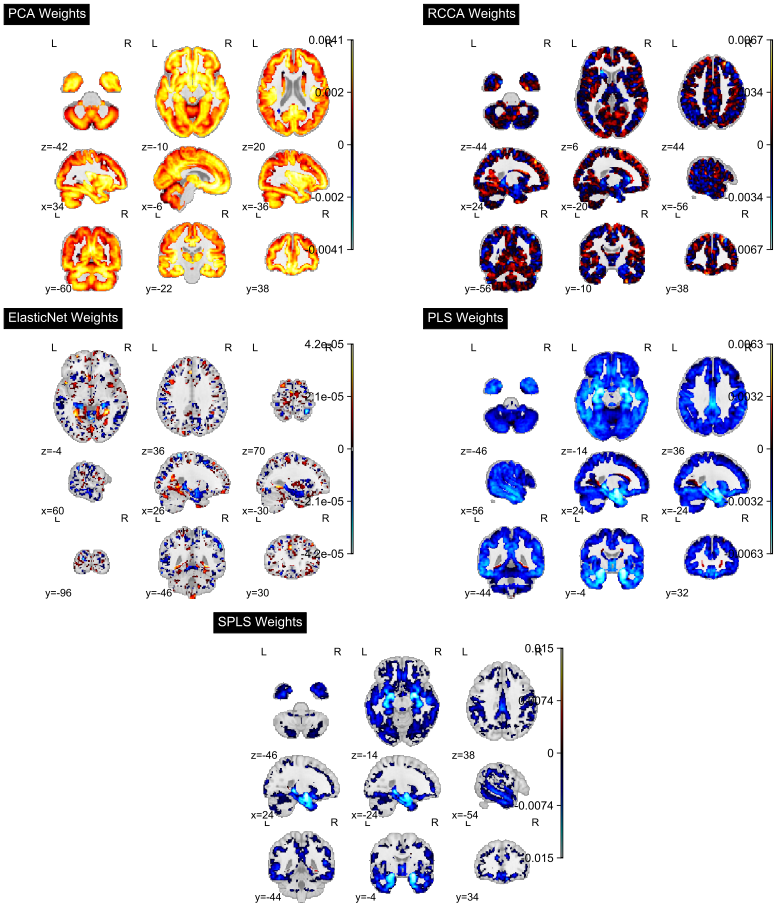


Figure III.9: ADNI: Statistical maps of brain structure weights for each model.

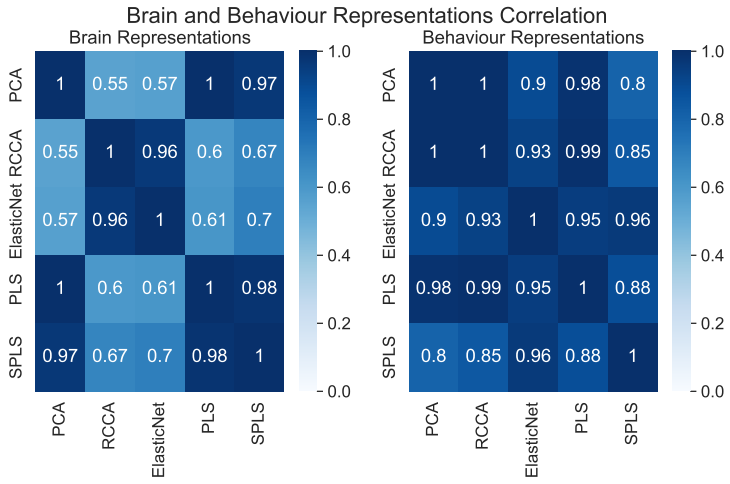


Figure III.10: ADNI: Correlation between the brain and behaviour representations for each model.

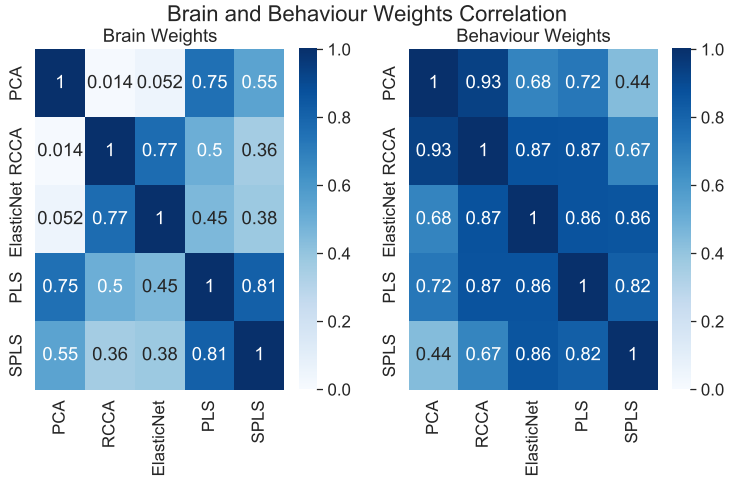


Figure III.11: ADNI: Correlation between the brain and behaviour weights for each model.

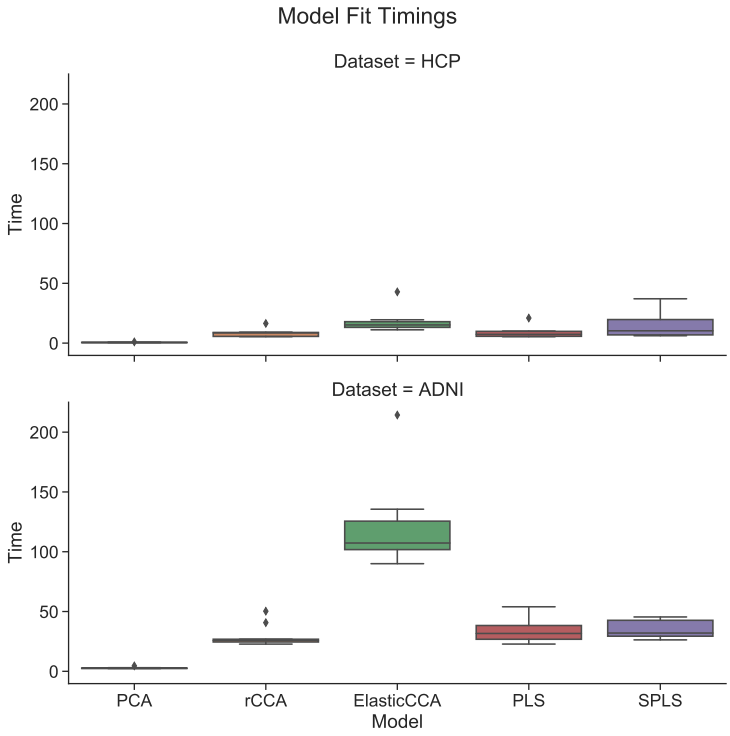


Figure III.12: Time taken to fit each model over ten runs. The interquartile range is plotted as a box with whiskers drawn to the farthest datapoint within 1.5 times the interquartile range.

Chapter IV

Insights From Generating Simulated Data for CCA

Contents

Preface

This chapter, deriving insights from various projects, lays out both my arguments for the use of loadings in the interpretation of CCA models and a number of computational tricks that we used to generate simulated data with significantly higher dimensions than have been previously considered in the literature. The simulated data generation methods were used to generate simulated data in Mihalik, Chapman, Rick A Adams, et al. (2022a). The arguments for the use of loadings influenced our choice of loadings for the interpretation of the results in Rick A. Adams et al. (2024).

1 Introduction

Despite its popularity, there is an ongoing debate in the CCA literature regarding the interpretation of model weights versus loadings (Gu and Wu, 2018). This chapter aims to contribute to this debate by providing mathematical insights from generative models of CCA and empirical results from simulated data with higher dimensionality than previously considered in the literature.

We begin by categorizing methods for generating CCA simulated data into explicit and implicit latent variable models. This categorization allows us to compare and contrast the generative models in CCA literature with the generative model for linear regression. We highlight that in linear regression, regularization can be interpreted as a prior on the weights, whereas in CCA, it is perhaps more natural to interpret regularization as a prior on the loadings. By leveraging computational tricks, we demonstrate how to generate simulated data with significantly higher dimensions than previously considered in the literature (Helmer et al., 2020; Matkovič et al., 2023).

Furthermore, we rigorously prove that loadings are invariant to columnwise transformations in data matrices, unlike weights. This property makes CCA unique compared to Principal Component Analysis (PCA) or Partial Least Squares (PLS) and is particularly relevant in fields like brain-behavior studies, where data preprocessing often involves columnwise manipulation.

Our experimental design focuses on two main aspects. First, we evaluate the ability of CCA models to accurately recover the true model weights and loadings. Second, we examine the out-of-sample performance, which is often observed to be poor in practical datasets despite statistical significance, particularly for PLS-based models. This observation led us to question whether the issue lies in poor model fit or a lack of signal in the data with weak or biologically spurious correlations.

One of our most striking findings, consistent with the previous chapter, is the efficacy of Ridge Regularized CCA models compared to PLS models in identifying high correlations under anisotropic noise conditions. This complements earlier work (Helmer et al., 2020) that found that the number of samples needed to find high correlations increases with dimensionality; our results suggest that the important variable is the dimensionality of the smaller view.

Through this chapter, we aim to provide a comprehensive understanding of the relationship between weights and loadings in CCA models, the impact of regularization on model interpretation, and the performance of CCA models in high-dimensional settings. By unifying generative perspectives, proving mathematical properties, and conducting extensive simulations, we contribute to the ongoing debate in the CCA literature and provide valuable insights for researchers and practitioners applying CCA in various domains.

2 Background: Weights and Loadings in Canonical Correlation Analysis

As discussed in chapter II, Canonical Correlation Analysis (CCA) is a powerful multivariate statistical technique that explores the relationships between two sets of variables. It can be interpreted in two ways: either as a method that finds linear combinations of variables in two datasets that exhibit the highest correlation, or as a technique that estimates latent variables that are maximally correlated.

The concept of latent variables is particularly important in biomedical applications, as it can help uncover underlying factors influencing observable data. For example, in brain-behavior studies, latent variables may represent hidden neurological or cognitive processes that drive the relationship between brain structure or function and behavioral outcomes. Similarly, in imaging-genetics, latent variables can capture the genetic factors that influence brain morphology or activity patterns. By introducing latent variables, CCA enables researchers to gain a deeper understanding of complex phenomena like gene expression, pathologies, and normative variations in health-related data (Lawry Aguila, Chapman, and Altmann, 2023).

CCA's practical application revolves around two main approaches: the discriminative approach and the generative approach. The discriminative approach, represented as the 'backward model' in Figure IV.1, uses weights to estimate highly correlated latent variables from observed data. It focuses on modeling the conditional distribution of the latent variables given the observed data, denoted as $P(Z|X^{(1)}, X^{(2)})$. In contrast, the generative approach, known as the 'forward model', emphasizes the data generation process and employs loadings to describe the relationship between latent variables and observed data. It models the joint distribution of the observed data conditioned on the latent variables, expressed as $P(X^{(1)}, X^{(2)}|Z)$.

The distinction between the generative and discriminative approaches in CCA is analogous to the different interpretations of Principal Component Analysis (PCA) (Park, Ceulemans, and Van Deun, 2023). PCA can also be viewed from a generative perspective, where the observed data are assumed to be generated from latent variables (Tipping and Bishop, 1999), or from a discriminative perspective, where the principal components are linear combinations of the observed variables that maximize the variance (Hotelling, 1933).

In CCA research, there is an ongoing debate regarding the interpretation of models in terms of weights or loadings (Gu and Wu, 2018). Weights are often

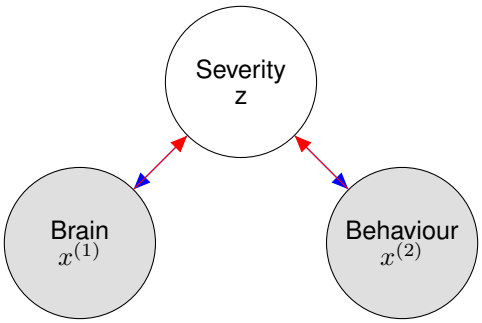


Figure IV.1: ***Forward and Backward Multiview Models:*** The generative/forward and discriminative/backward approaches in CCA.

preferred for prediction tasks, as they directly relate the observed variables to the latent variables. On the other hand, loadings are favored for interpretation, as they provide insights into the structure and relationships within the data (Z. Liu et al., 2022). This discussion is particularly relevant to our work in chapter III and various studies involving sparse CCA and sparse Partial Least Squares (PLS), where understanding the meaning and implications of sparse loadings and weights is crucial.

Given the importance of this topic, especially in the context of our work in chapter III and other studies employing variants of sparse CCA and sparse PLS, it is essential to delve deeper into the interpretation of sparse loadings and weights. In the following sections, we will explore the mathematical properties and practical implications of weights and loadings in CCA, with a focus on sparse and regularized models. By addressing this key aspect of CCA, we aim to contribute to the ongoing debate and provide insights that can guide the application and interpretation of CCA in various domains, including neuroimaging, genetics, and health-related research.

3 Unifying Generative Perspectives in CCA: Explicit and Implicit Latent Variable Models

This section categorizes the generative models in CCA literature into explicit and implicit latent variable types, each offering distinct insights into the data generation process and the relationship between weights and loadings.

3.1 Additional Notational Conventions

We will use some additional notational convention to describe probabilistic models. We will use lowercase letters to represent samples from a distribution, and uppercase letters to represent random variables. For example, x represents a sample from the distribution $P(X)$, and X represents the random variable X. We will use $\sim$ to denote the sampling process, and $|$ to denote conditioning. For example, $x|z \sim \mathcal{N}(\mu, \Psi)$ represents a sample x from a Gaussian distribution with mean Wz and covariance Ψ conditioned on the latent variable z. We also introduce the notation $w_j^{(i)}$ to refer to the loading of the j-th feature in the i-th view on a latent variable, as well as $W^{(i)}$ to refer to the matrix of loadings for the i-th view on all latent variables.

3.2 Explicit Latent Variable Models: Probabilistic CCA and GFA

In explicit latent variable models, we assume each view is generated from a linear model with added noise, conditional on the latent variable. The distributions for the two views are given by:

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

$$x^{(i)} \sim \mathcal{N}(W^{(i)}z + \mu^{(i)}, \Psi^{(1)}) \tag{IV.2}$$

Where z represents a sample from the latent guassian distribution, $x^{(i)}$ represents a sample from the i-th view, $W^{(i)}$ represents the model loadings, $\mu^{(i)}$ represents the mean, and Ψ_i represents the noise covariance matrix for the i-th view.

Bach and Jordan (2005) established that the maximum likelihood solution of this model relates the CCA weights to the loadings by the within-view covariance Σ_{ii}:

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

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

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

We estimate the covariance matrices Σ_{ii} from the data using sample covariance matrices $\hat{\Sigma}_{ii}$. For Identity covariance matrices, the CCA weights are the same as the loadings.

In the Group Factor Analysis (GFA) model, we assume diagonal covariance. This assumption enhances computational efficiency and supports extensions like sparsity on loadings:

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

$$x^{(i)} \sim \mathcal{N}(W^{(i)}z, {\sigma_i}^2 I) \tag{IV.6}$$

The joint distribution of the two views with an explicit latent variable model is given by (Bach and Jordan, 2005):

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

Where $\Psi^{(i)}$ is the noise covariance matrix for the i-th view. 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-th view is given by:

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

When $W^{(i)}W^{(i)\,T}$ has large eigenvalues as compared to $\Psi^{(i)}$, Σ_{ii} is often called a 'spiked covariance matrix' (Johnstone, 2001).

3.2.1 Aligning Probabilistic PCA and GFA

Probabilistic PCA (Tipping and Bishop, 1999) is a generative model that assumes the data is generated from a linear model with added noise:

$$x \sim \mathcal{N}(Wz, \sigma^2 I) \tag{IV.9}$$

GFA can thus be seen as a generalization of Probabilistic PCA, where the noise covariance matrix is diagonal and the latent variable is shared across multiple views. An interesting consequence of this is that it tells us that for low levels of isotropic noise, we ought to be able to recover latent variables with just one view. For this reason, in my opinion, PCA should always be used as a baseline in CCA studies.

3.3 Implicit Latent Variable Models: The Joint Covariance Matrix Perspective

The joint covariance matrix perspective, prevalent in sparse CCA literature (Suo et al., 2017; M. Chen et al., 2013), emphasizes covariance matrices over direct modeling of latent variables. This approach allows us to directly control the sparsity of the weights and the strength of the canonical correlations by constructing the covariance matrices accordingly. By focusing on the covariance structure, we can generate data with desired properties without explicitly modeling the latent variables. This is achieved by constructing the joint covariance matrix of the distribution $P(X^{(1)}, X^{(2)})$:

$$\begin{bmatrix} X^{(1)} \\ X^{(2)} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right) \tag{IV.10}$$

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

For clarity and simplicity in our discussion, we refer to a single canonical correlation coefficient, ρ, without loss of generality. This allows us to focus on the structure of the covariance matrices without the complexity of multiple canonical correlations.

In constructing the between-view covariance matrices Σ_{12} and Σ_{21}, we control the true signal by setting active variables and correlations. Specifically, the between-view covariance matrix is constructed as follows:

$$\Sigma_{12} = \rho \Sigma_{11} u_1^{(1)} u_1^{(2)\,T} \Sigma_{22} \tag{IV.11}$$

Here, ρ is the canonical correlation, and $u_1^{(i)}$ is the first column of the matrix of

weights $U^{(i)}$ for the i-th view.

This perspective simplifies the structure of covariance matrices, focusing on the relationship between views as controlled by the canonical correlation coefficient, ρ, and the weights $u^{(i)}$.

3.4 Summary of Data Generation Methods

To summarize the key differences between the data generation methods discussed above, we present two tables. Table 1 compares the covariance structures of each method, highlighting how the within-view and between-view covariances are modeled. Table 2 illustrates the relationship between weights and loadings in both population and sample cases, emphasizing the implications for sparsity and identifiability.

Table 3.1: Covariance Structures in Data Generation Methods

	Method	**Within-view Covariance $\boldsymbol{\Sigma_{ii}}$**	**Between-view Covariance $\boldsymbol{\Sigma_{12}}$**
Explicit	Probabilistic CCA	$W^{(i)}W^{(i)\,T} + \Psi^{(i)}$	$W^{(1)}W^{(2)\,T}$
	GFA	$W^{(i)}W^{(i)\,T} + {\sigma^{(i)}}^2 I$	$W^{(1)}W^{(2)\,T}$
Implicit	Joint Covariance	Σ_{ii}	$\rho\Sigma_{11}u_1^{(1)}u_1^{(2)\,T}\Sigma_{22}$
	Joint Covariance (Identity)	I	$\rho u_1^{(1)}u_1^{(2)\,T}$

Table 3.2 summarizes 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 Σ is identity, but the sample covariance matrix $\hat{\Sigma}$ is only an approximation. An important observation is that for the implicit latent variable models, we can generate data with sparse weights but not, in general, sparse loadings. For the explicit latent variable models, we can generate data with sparse loadings but not, in general, sparse weights.

Table 3.2: Relationship Between Weights and Loadings in Population and Sample Cases

	Method	Case	Weights	Loadings
Explicit	Probabilistic CCA	Population	$(W^{(i)}W^{(i)\,T} + \Psi^{(i)})^{-1}W^{(i)}$	$W^{(i)}$
		Sample	$\hat{\Sigma_{ii}}^{-1}W^{(i)}$	$W^{(i)}$
	GFA	Population	$(W^{(i)}W^{(i)\,T} + {\sigma^{(i)}}^2 I)^{-1}W^{(i)}$	$W^{(i)}$
		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	$U^{(i)}$	$\hat{\Sigma_{ii}}\hat{U^{(i)}}$

3.5 Regularization and Generative Models

Regularization is crucial in CCA to prevent overfitting and promote interpretability. However, the way regularization is interpreted in CCA differs from linear regression due to the latent variable nature of CCA models. In linear regression, regularization can be directly interpreted as a prior on the weights. In contrast, for CCA, regularization can be interpreted as a prior on either the loadings or the weights, depending on the generative perspective. This distinction has important implications for model interpretation and the identifiability of weights in CCA.

3.5.1 Regularization and the Generative Model for Linear Regression

Linear regression assumes data generation from a linear model with added noise:

$$y = xU + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I) \tag{IV.12}$$

Here, y are samples of the target variable, x samples from the data matrix, U the regression coefficients, and ϵ represents independent and identically distributed (i.i.d.) Gaussian noise.

Lasso Regression The Lasso imposes a Laplace prior on the regression coefficients, leading to a double-exponential prior on weights:

$$U \sim \mathcal{L}(0, \lambda) \tag{IV.13}$$

Ridge Regression Ridge regression, in contrast, employs a Gaussian prior on the regression coefficients, equivalent to a Gaussian prior on weights:

$$U \sim \mathcal{N}(0, \lambda) \tag{IV.14}$$

3.5.2 Regularization and Generative Models for CCA

CCA models differ in their approach to regularization compared to linear regression because they are latent variable models.

Explicit Latent Variable Model Regularization in the context of the explicit latent variable naturally relates to priors on the loadings $W^{(i)}$. For example, sparsity in the loadings can be achieved by imposing a Laplace prior on the loadings:

$$W^{(i)} \sim \mathcal{L}(0, \lambda) \tag{IV.15}$$

This expresses the prior belief that latent factors only explain the data through a small number of features. For example, in the context of latent factors in brain-behavior studies, this prior belief is equivalent to the assumption that a latent mode of variance (perhaps a subtype) is only expressed through a small number of brain regions.

Implicit Latent Variable Model In the implicit latent variable model of CCA, the joint likelihood is modeled as a block covariance matrix Σ(Suo et al., 2017), constructed from the weights $U^{(i)}$.

$$\Sigma = \begin{bmatrix} \Sigma_1 & \Sigma_1 U^{(1)} \rho U^{(2)\,T} \Sigma_2 \\ \Sigma_2 U^{(2)} \rho U^{(1)\,T} \Sigma_1 & \Sigma_2 \end{bmatrix} \tag{IV.16}$$

Where the off-diagonal blocks $\Sigma_1 U^{(1)} \rho U^{(2)\,T} \Sigma_2$ and its transpose represent the between-view covariance matrices. These matrices are functions of the weights $U^{(i)}$ and within-view covariance matrices Σ_i, modulated by ρ, the canonical correlation coefficients.

Here the regularization naturally relates to priors on the weights $U^{(i)}$. For example, sparsity in the weights can be achieved by imposing a Laplace prior on the weights:

$$U^{(i)} \sim \mathcal{L}(0, \lambda) \tag{IV.17}$$

This expresses the more nuanced prior belief that the latent factors are expressed through a subset of features and then distorted by arbitrary rotations as well as the within-view covariance matrices. Manipulating equation IV.3, the conditional distribution of the implicit latent variable model we have:

$$x^{(i)}|z \sim \mathcal{N}(\Sigma_i U^{(i)} R z = W^{(i)} z, \Sigma_i - W^{(i)} W^{(i)\,T} = \Psi^{(i)}) \tag{IV.18}$$

$$z \sim \mathcal{N}(0, I) \tag{IV.19}$$

The arbitrary rotation matrix R means that for multidimensional $U^{(i)}$, even if $\Sigma_i = I$, and even if the true loadings are sparse, the weights may still not be sparse!

$$x^{(i)}|z \sim \mathcal{N}(U^{(i)} R z = W^{(i)}_{sparse} z, \Sigma_i - W^{(i)}_{sparse} W^{(i)\,T}_{sparse} = \Psi^{(i)}) \tag{IV.20}$$

$$z \sim \mathcal{N}(0, I) \tag{IV.21}$$

Alternatively, even if we know the true weights (i.e. $R = I$), the CCA model may not be able to recover them. This is to say they are not, in general, identifiable (Park, Ceulemans, and Van Deun, 2023). In other words there are multiple values of $W^{(i)}$ that can produce the same covariance structure.

We can illustrate this with a trivial example:

$$\Sigma_1 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.5 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{IV.22}$$

(IV.23)

In this example, we show that the same covariance matrix Σ_1 can be obtained using different weight matrices. The first weight matrix has entries [1, 0, 1], while the second weight matrix has entries [0.5, 0.5, 1]. This example clearly demonstrates the non-identifiability issue in the implicit latent variable model, where multiple weight matrices can produce the same covariance structure. This means that even if we know the true covariance structure, we may not be able to uniquely recover the true weights.

One practical implication of this observation is that it raises serious questions about using stability selection, a common practice in the sparse CCA literature (Mihalik, Ferreira, Moutoussis, et al., 2020; Deng et al., 2021), to select the optimal regularization parameter.For instance, suppose we run stability selection multiple times on the same dataset to select the optimal regularization parameter. Due to the non-identifiability of weights, each run may result in different rotations of the weights, even though the underlying representations and correlations remain the same. This can lead to inconsistent selection of the regularization parameter across runs, potentially resulting in suboptimal hyperparameter choices or incorrect conclusions about the sparsity structure of the data.

In summary, understanding the generative perspectives in CCA is crucial for interpreting regularization, sparsity, and identifiability in these models. The explicit latent variable model allows for intuitive priors on the loadings, while the implicit latent variable model enables priors on the weights, albeit with less straightforward interpretations. The non-identifiability issue in the implicit model highlights the challenges in recovering unique weights and raises questions about the reliability of stability selection. By considering these generative perspectives, researchers

can make more informed choices when applying regularization and interpreting the results of CCA models.

4 Invariance of Loadings in CCA: An Intuitive Mathematical Argument

In this section, we present an intuitive mathematical argument for favoring loadings over weights in the interpretation of CCA models. We will demonstrate that loadings are invariant to certain common transformations of the data matrix, including scaling, duplication, and summation of columns. This property is not shared by weights. This invariance has significant practical implications, especially when working with heterogeneous or transformed data.

4.1 Solving CCA in Principal Component Space

Consider the singular value decomposition (SVD) of the data matrices:

$$X^{(i)} = U^{(i)} S^{(i)} V^{(i)\,T} \tag{IV.24}$$

Here, $U^{(i)}$ contains the left singular vectors (principal components) of $X^{(i)}$, $S^{(i)}$ is a diagonal matrix of singular values, and $V^{(i)}$ contains the right singular vectors. The columns of $U^{(i)}$ span the column space of $X^{(i)}$, which is the space of all possible linear combinations of the columns of $X^{(i)}$. Intuitively, the column space captures all the directions in which the data varies.

The CCA objective is to find weights $u^{(1)}, u^{(2)}$ that maximize the correlation between the canonical variables $X^{(1)}u^{(1)}$ and $X^{(2)}u^{(2)}$:

$$\max_{u^{(1)},u^{(2)}} \mathrm{Corr}(X^{(1)}u^{(1)}, X^{(2)}u^{(2)}) = \max_{u^{(1)},u^{(2)}} \mathrm{Corr}(U^{(1)}S^{(1)}V^{(1)\,T}u^{(1)}, U^{(2)}S^{(2)}V^{(2)\,T}u^{(2)}) \tag{IV.25}$$

By reparameterizing the weights as $v^{(i)} = S^{(i)}V^{(i)\,T}u^{(i)}$, we obtain:

$$\max_{v^{(1)},v^{(2)}} \mathrm{Corr}(U^{(1)}v^{(1)}, U^{(2)}v^{(2)}) \tag{IV.26}$$

This shows that CCA can be solved entirely in the principal component space spanned by the matrices $U^{(i)}$. The loadings $w_j^{(i)}$, defined as the correlations between the original features $X_j^{(i)}$ and the canonical variables $U^{(i)}v^{(i)}$, capture the relationships in this space.

4.2 Invariance of Loadings to Data Transformations

We now show that the loadings are invariant to certain transformations of the data matrix $X^{(i)}$, while the weights are not. The key insight is that these transformations change the right singular vectors $V^{(i)}$ and singular values $S^{(i)}$, but not the left singular vectors $U^{(i)}$. Since the loadings depend only on $U^{(i)}$, they remain invariant. Moreover, these transformations preserve the column space of $X^{(i)}$, which is why the principal components $U^{(i)}$ are unaffected.

4.2.1 Scaling Transformation

Consider a diagonal scaling matrix B that scales the columns of $X^{(i)}$:

$$\tilde{X}^{(i)} = X^{(i)}B = (U^{(i)}S^{(i)}V^{(i)\,T})B = U^{(i)}(S^{(i)}B)(V^{(i)\,T}) \tag{IV.27}$$

The weights $\tilde{u}^{(i)}$ in the transformed space are related to the original weights by $\tilde{u}^{(i)} = B^{-1}u^{(i)}$, and thus change with the scaling. However, the principal components $U^{(i)}$ remain unchanged, so the loadings $\tilde{w}_j^{(i)} = \mathrm{Corr}(\tilde{X}_j^{(i)}, U^{(i)}v^{(i)}) = w_j^{(i)}$ are invariant.

4.2.2 Duplication Transformation

Consider duplicating columns of $X^{(i)}$ using a transformation matrix B that contains an identity matrix I_n and a duplication matrix D:

$$B = \begin{bmatrix} \\ I_n \; D \end{bmatrix}, \quad \tilde{X}^{(i)} = X^{(i)}B = U^{(i)}(S^{(i)}B)(V^{(i)\,T}) \tag{IV.28}$$

The weights $\tilde{u}^{(i)}$ become underdetermined in the transformed space due to the added linear dependencies. However, since the column space of $\tilde{X}^{(i)}$ is the same as that of $X^{(i)}$, the principal components $U^{(i)}$ and thus the loadings remain unchanged.

4.2.3 Linear Combination Transformation

Consider adding or removing linear combinations of columns using a transformation matrix B that contains an identity matrix I_n and a coefficient matrix C:

$$B = \begin{bmatrix} I_n & C \end{bmatrix}, \quad \tilde{X}^{(i)} = X^{(i)}B = U^{(i)}(S^{(i)}B)(V^{(i)\,T}) \tag{IV.29}$$

As before, the weights $\tilde{u}^{(i)}$ change in the transformed space, but since the column space is preserved, the principal components $U^{(i)}$ and loadings remain invariant.

4.3 Practical Implications

The invariance of loadings to data transformations has significant practical implications, especially in fields like biomedical research, psychometrics, and social sciences where questionnaire and survey data are common:

- **Interpretability**: Loadings provide a consistent interpretation of the relationships between the original features and the canonical variables, even if the data is rescaled or transformed. This is particularly valuable in interdisciplinary research, where different data normalization practices may be employed.
- **Feature Selection**: Decisions about including, excluding, or combining features can be made based on the loadings without worrying about their impact on the CCA solution. This is especially relevant when dealing with summary measures that effectively sum other variables, a common scenario in questionnaire and biomedical data. The invariance of loadings to such alterations in the data structure makes them a more robust choice for interpreting relationships between variables in these contexts.
- **Robustness**: CCA models can be trained on transformed data (e.g., normalized or standardized) while still allowing for meaningful interpretation in the original feature space. While the identifiability of weights can be partially solved by the standardization of data, and while this is a common practice, it is not always necessary or desirable and always introduces assumptions.

In conclusion, this section provides a strong mathematical foundation for the preference of loadings over weights in the interpretation of CCA models. The in-

variance of loadings to columnwise transformations, including scaling and linear combinations, ensures a more robust and consistent interpretation of variable relationships. This property is especially valuable in fields dealing with heterogeneous or transformed data, where data preprocessing choices may vary. By focusing on loadings, researchers can obtain more reliable insights into the underlying structure of their data, facilitating cross-disciplinary collaborations and the advancement of knowledge.

5 Efficient Sampling of Simulated CCA Data

Efficient sampling is crucial for CCA because it allows researchers to work with larger datasets and explore more complex or more nuanced relationships between variables, ultimately expanding the scope of research and analysis. Traditional methods can be computationally intensive and storage-demanding, especially for large datasets. This has in practice limited the dimensionality of simulated data, restricting the scope of research and analysis. For example Matkovic et al. (2023) simulate data with 8,000 observations and 100 features while Helmer et al. (2020) used at most 10,000 observations and 64 features. We were interested in the behavior of CCA in high-dimensional settings like voxel-wise MRI and connectivities, which can have hundreds of thousands of features (Jack Jr et al., 2008) and up to tens of thousands of observations (Sudlow et al., 2015). By leveraging the assumptions that biomedical data often exhibit low-rank and/or sparse covariance structures, we develop efficient sampling methods that overcome the computational and storage limitations associated with high-dimensional data.

5.1 Challenges with High-Dimensional Data

Direct sampling from a multivariate normal distribution is impractically slow for high-dimensional data, which has been a core research challenge for Monte Carlo methods (Mackay, 1998). The implicit latent variable model, in particular, requires storage of the full covariance matrix, which is prohibitive for high-dimensional data. For example, a covariance matrix with 100,000 dimensions would require 80GB of memory, far exceeding the capacity of most personal computers.

5.2 Efficient Sampling for Explicit Latent Variable Models

The explicit latent variable model offers more efficient approaches for sampling high-dimensional data by employing sparse and low-rank covariance matrices.

5.2.1 Sampling from Multivariate Normal Distributions

An efficient approach to sampling from a multivariate normal distribution is to use the Singular Value Decomposition (SVD) or Cholesky decomposition of the covariance matrix. This involves decomposing the covariance matrix and using the resulting components to transform samples from a standard multivariate normal distribution:

$$Z \sim \mathcal{N}(0, I) \tag{IV.30}$$

$$X = \Sigma^{1/2} Z \tag{IV.31}$$

Where $\Sigma^{1/2}$ is a square root of the covariance matrix, obtained through SVD or Cholesky decomposition. This is the same as the generative model for the explicit latent variable model, where $\Sigma^{1/2}$ is the matrix of loadings. Low-rank noise can be added by sampling from an independent multivariate normal distribution and adding it to the transformed samples. This approach requires sampling from a univariate normal distribution and performing a matrix multiplication of complexity $\mathcal{O}(np^2)$.

5.2.2 Using Sparse and Low-Rank Covariance Matrices

Sparse covariance matrices, with many zero entries, reduce both computational complexity and storage requirements. For example, a sparse covariance matrix with 100,000 dimensions and 10% density would only require 8GB of memory to store.

Low-rank covariance matrices further reduce complexity by storing only the factorized rank-k components, reducing storage requirements to $\mathcal{O}(kp)$. For example, a low-rank covariance matrix with 100,000 dimensions, 10% density, and rank 1000 would only require 80MB of memory to store. This approach also requires drawing $\mathcal{O}(kp)$ samples from a univariate normal distribution and performing a matrix multiplication with complexity $\mathcal{O}(nkp)$, rather than $\mathcal{O}(np^2)$ for the full-rank case.

5.3 Calculating True Canonical Correlations and Weights

The population canonical correlations can be controlled by varying the signal-to-noise ratio (SNR), i.e., the ratio of the signal variance to the noise variance.

For the explicit latent variable model, the loadings are obtained directly as the low-rank square root of the covariance matrix. The weights can be calculated from the loadings and the covariance matrix using the relationship:

$$\hat{W}^{(i)} = \Sigma_{ii}\hat{U}^{(i)}R \tag{IV.32}$$

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

$$\hat{U}^{(i)}R = \Sigma_{ii}^{-1}\hat{W}^{(i)} \tag{IV.33}$$

Although inverting the $\mathcal{O}(p^2)$ covariance matrix is computationally expensive, the Sherman-Morrison-Woodbury formula can be used to calculate the inverse in $\mathcal{O}(kp^2)$ time for a rank-k covariance matrix. This allows for the calculation of weights in $\mathcal{O}(kp^2)$ time, which is faster than the $\mathcal{O}(p^3)$ time required to calculate the weights directly from the covariance matrix.

In the next section, we will present experiments demonstrating the relationship between weights and loadings in simulated data using these efficient sampling techniques.

6 Experiment Design

Our goal in this section is to empirically demonstrate the relationship between weights and loadings in CCA models as well as to better understand the behavior of CCA models in the high-dimensional settings that section 5 enables, and which are of interest in the neuroimaging community.

The first set of experiments illustrates the relationship between weights and loadings in simulated data using explicit latent variable models with identity and non-identity covariance matrices. The second set of experiments illustrates the ammount of information that can be recovered from simulated data using CCA and PLS models with varying signal-to-noise ratios and sample sizes.

6.1 Exploring the Relationship Between Weights and Loadings in CCA Using Simulated Data

Our first experiment is designed to illustrate the challenges of recovering the true weights and loadings respectively in CCA models for explicit and implicit latent variable models with identity and non-identity covariance matrices.

We compare the true weights derived from the data generation model with the estimated weights of CCA, Ridge CCA, Elastic Net, PLS, and PCA models. We expect that when the covariance matrix is identity, the weights and loadings will be identical. When the covariance matrix is non-identity, we expect that the weights and loadings will be different. Moreover, we expect that the estimated loadings will be more stable than the estimated weights for CCA models because the weights are not always identifiable. Under the explicit latent variable model, we expect that the weights will only be (close to) sparse when the covariance matrix is close to identity. This means we do not expect the Elastic Net model to improve on the Ridge CCA model since the Lasso regularizes the weights but not the loadings. Finally, we expect that when using the explicit latent variable model, for high signal-to-noise ratios, the PLS and even PCA models will recover the true weights and loadings because the majority of the variance is explained by the latent variables.

6.1.1 Detailed Parameters of Simulated Data for Weights and Loadings Analysis in CCA

We generate data with 100 samples and 10 features in each view. We then generate data under two implicit latent variable models and two explicit latent variable models. The ridge penalty is coarsely tuned between 0.1 and 0.9 in order to illustrate the effect of regularization as we already show the corner cases of no regularization (CCA) and full regularization (PLS). For the Elastic Net model, we tune the l1 ratio between 0.1 and 0.9. This ensures that the Elastic Net has some sparsity as compared to the Ridge model, effectively avoiding the corner case of no sparsity where the Elastic Net is equivalent to the Ridge model. We summarize the parameters of these experiments in table 6.1.

6.2 Assessing Information Recovery in CCA and PLS Models Under Varying Signal-to-Noise Ratios

Our next experiment was motivated by the observation that PLS models (including sparse PLS) often exhibit low but non-zero out of sample correlations in real high-

Table 6.1: Simulated Data Parameters for Weight and Loadings Recovery Experiments

Parameter	**Value**
Number of samples (n)	100 train, 500 test
Number of features in View 1 (p)	10
Number of features in View 2 (q)	10
True Latent dimensions	1
Fraction of active features View 1	0.5
Fraction of active features View 2	0.5

dimensional data. We want to understand how much of this is due to the fact that PLS models optimize covariance rather than correlation, and how much is due to the fact that the signal-to-noise ratio is too low. In order to understand this, we simulated data with varying signal-to-noise ratios and compared the out of sample correlations of PLS models with the out of sample correlations of Ridge CCA models with varying regularization. Since we are interested in studying these effects in high-dimensional data, we aimed to simulate data with similar numbers of features to real brain-behavior datasets. This means that we are only able to use our memory-efficient sampling methods for the explicit latent variable model.

6.2.1 Detailed Parameters of Simulated Data for Signal-to-Noise Simulations

We simulated data with 1000 samples and between 100 and 10,000 features in one view and 100 features in the other. These are of the same order of magnitude as typical brain-behaviour datasets. We summarise these data properties in table 6.2.

Table 6.2: Simulated Data Parameters for Brain-Behaviour Simulations

Parameter	**Value**
Number of features in View 1 (p)	100-10000
Number of features in View 2 (q)	100-10000
True Latent dimensions	1
Fraction of active features View 1	1.0
Fraction of active features View 2	1.0
Signal-to-noise ratio	0.001-1

6.3 Methodology for Constructing Correlated Covariance Matrices in CCA Simulations

In both experiments, we construct correlated covariance matrices by generating a random matrix A with entries drawn from a uniform distribution between -1 and 1. We then construct the covariance matrix as $\Sigma = AA^\top$. This ensures that the covariance matrix is positive semi-definite and also tends to produce strong correlations.

We plot an example of the covariance matrices for correlated covariance matrices in both views in figure IV.2.

Recalling table 3.1, note that in the implicit latent variable models, these covariance matrices are precisely the population within-view covariance matrices. In the explicit latent variable models, these covariance matrices are just the covariance matrices of the noise to which we add the signal covariance matrices. Nonetheless, for strong enough noise, this process ensures that there are large correlations between features.

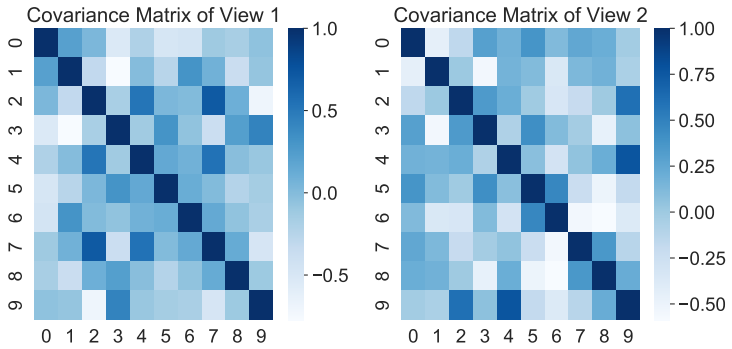


Figure IV.2: Example instances of correlated covariance matrices.

7 Experiment Results

7.1 Exploring the Relationship Between Weights and Loadings in CCA Using Simulated Data

We first present the results of the experiments demonstrating the relationship between weights and loadings in simulated data from explicit and implicit latent variable models with identity and non-identity covariance matrices.

For both cases, we plot the true weights and loadings along with the estimated weights and loadings for each model. We estimate model loadings by multiplying the model weights by the sample within-view covariance matrix following equation IV.3. This means that the estimated model loadings may not be sparse even when the estimated model weights are sparse and the *population* covariance matrix is identity.

We can also quantify the similarity between the true and estimated weights and loadings using the cosine similarity; a measure of the similarity between two vectors that is invariant to the scale of the vectors. The cosine similarity between two vectors is defined as the cosine of the angle between them (Luo et al., 2018). Since we are indifferent to the direction of the vectors, we take the absolute value of the cosine similarity. The absolute cosine similarity between two vectors is 1 if they are identical (up top a sign) and 0 if they are orthogonal.

7.1.1 Implicit Latent Variables (Sparse Weights)

Figure IV.3 shows the true and estimated weights and loadings for data generated from the implicit latent variable models with sparse weights. The Elastic net model

exhibits no false negatives (i.e. where the true weight is non-zero but the estimated weight is zero) in both cases. This shows that the Elastic Net model is able to recover the true weights and that the Lasso penalty is indeed inducing sparsity in the weights. The CCA model appears recover spectrum of the true weights much better for the identity covariance matrices than for the correlated covariance matrices. This is likely because the multicollinearity introduced makes the learnt weights substantially less stable with respect to a change in the data.

We plot the cosine similarity between the true and estimated weights and loadings for data generated from the implicit latent variable models with sparse weights in figure IV.4.

Interestingly, we see that for the identity covariance matrices, weight differences are smaller than loading differences. On the other hand for the correlated covariance matrices, the loading differences are smaller than the weight differences. This is evidence of the fact that the weights are not identifiable in the implicit latent variable model as suggested by our theory. Only when the covariance matrices are identity, and when there is only one latent variable, are the weights identifiable.

7.1.2 Explicit Latent Variables

Figure IV.5 shows the true and estimated weights and loadings for data generated from the explicit latent variable models with sparse loadings. The left column shows the results for the identity covariance matrices, while the right column shows the results for the correlated covariance matrices. Once again, the Elastic Net model exhibits no false negatives (i.e. where the true weight is non-zero but the estimated weight is zero) when the noise covariance matrix is identity such that both the weights and loadings are sparse.

Once again, we can quantify the similarity between the true and estimated weights and loadings using the cosine similarity (Figure IV.6).

Notably, when the noise covariance matrix is correlated, the difference in recovery of the weights is much larger than the difference in recovery of the loadings. Suprisingly, when the noise covariance matrix is identity, the PLS and PCA models appear to better recover the weights than the loadings in this case.

7.2 Assessing Information Recovery in CCA and PLS Models Under Varying Signal-to-Noise Ratios

In Figures IV.7 and IV.8 we plot the test correlation (score) varying the signal-to-noise ratio and the number of features under the identity and correlated noise covariance

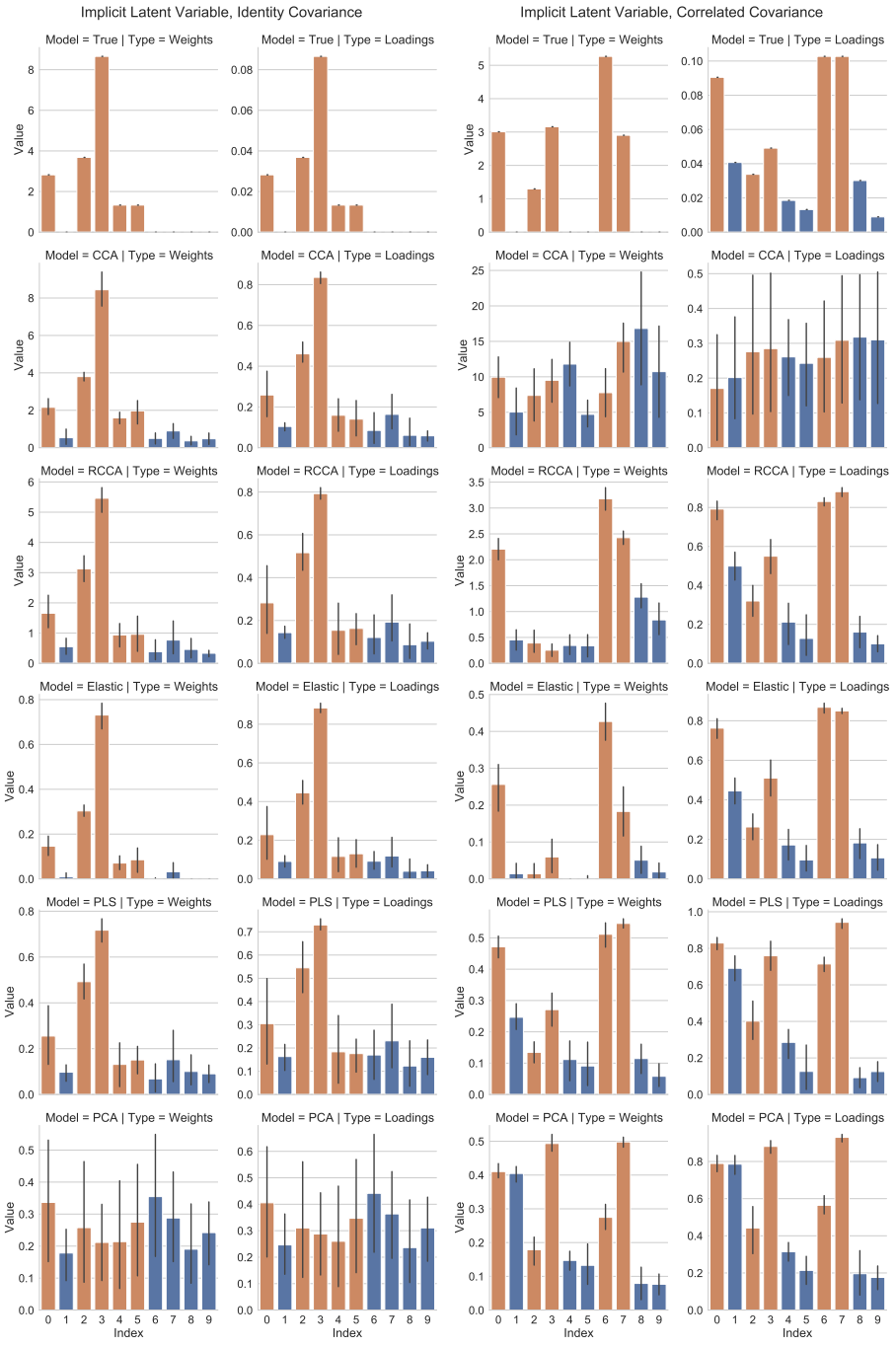


Figure IV.3: Bar plots of the true and estimated weights and loadings for data generated from the implicit latent variable models with sparse weights. The left column shows the results for the identity covariance matrices, while the right column shows the results for the correlated covariance matrices.

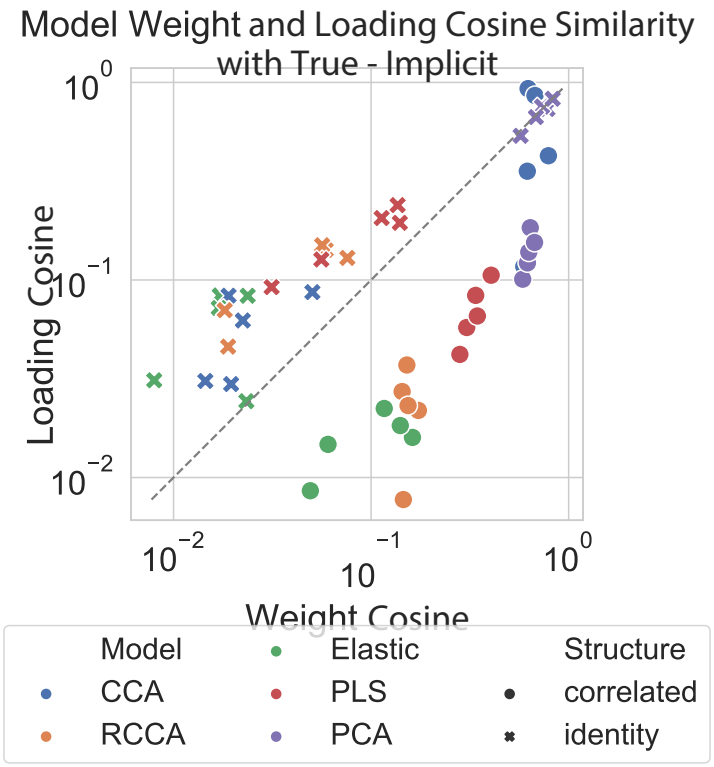


Figure IV.4: Cosine similarity between the true and estimated weights and loadings for data generated from the implicit latent variable models with sparse weights. We plot each run as a point on a scatter plot with a log scale. The grey line indicates where the similarity between weights and loadings are equal.

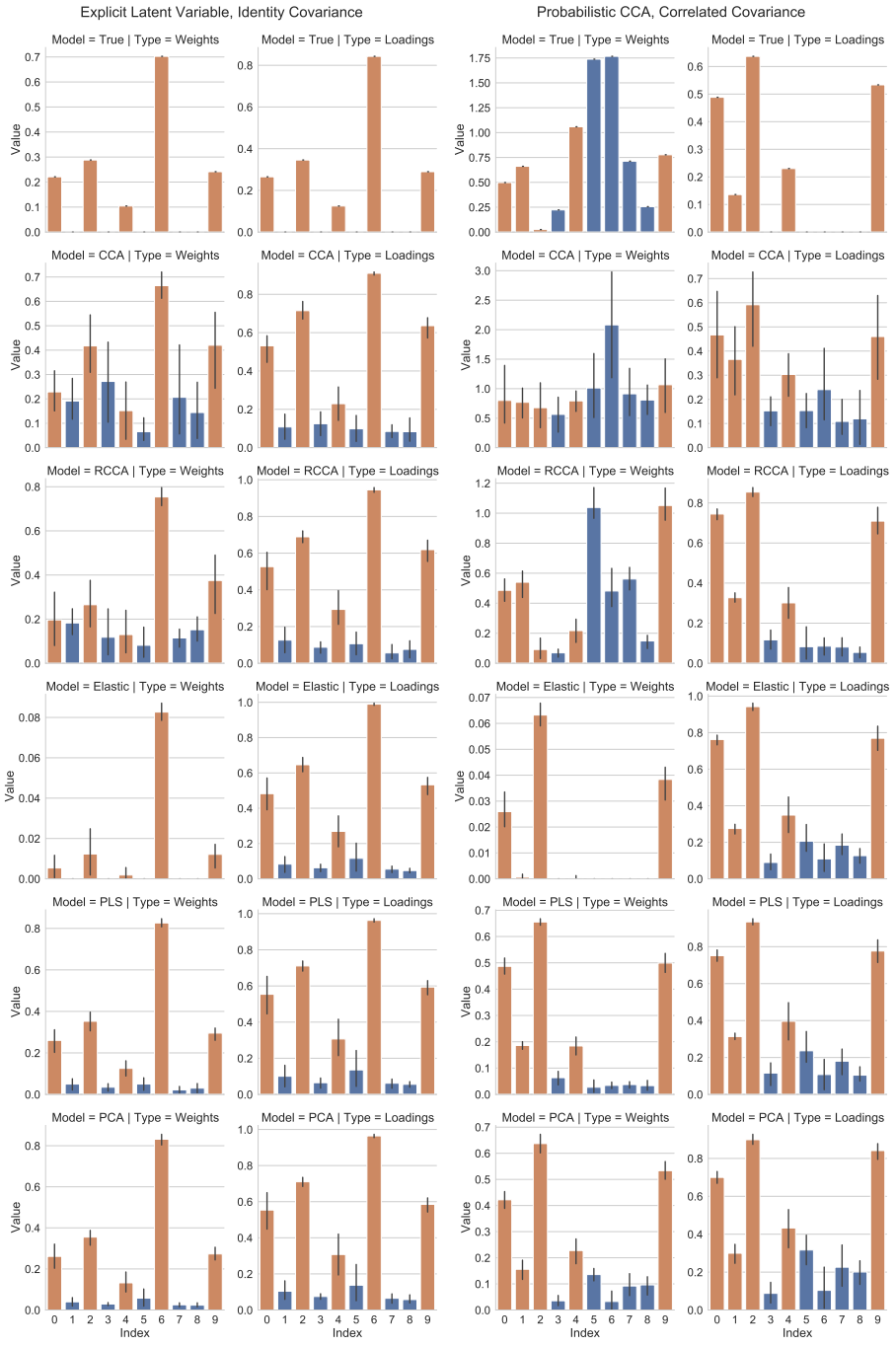


Figure IV.5: Bar plots of the true and estimated weights and loadings for data generated from the implicit latent variable models with sparse weights. The left column shows the results for the identity covariance matrices, while the right column shows the results for the correlated covariance matrices.

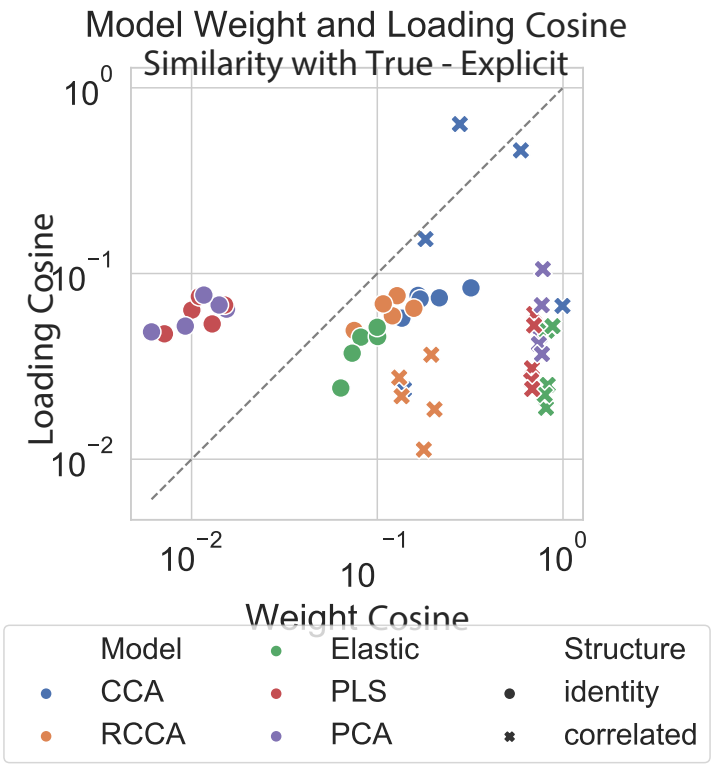


Figure IV.6: Cosine similarity between the true and estimated weights and loadings for data generated from the explicit latent variable models with sparse loadings. We plot each run as a point on a scatter plot with a log scale. The grey line indicates where the similarity between weights and loadings are equal.

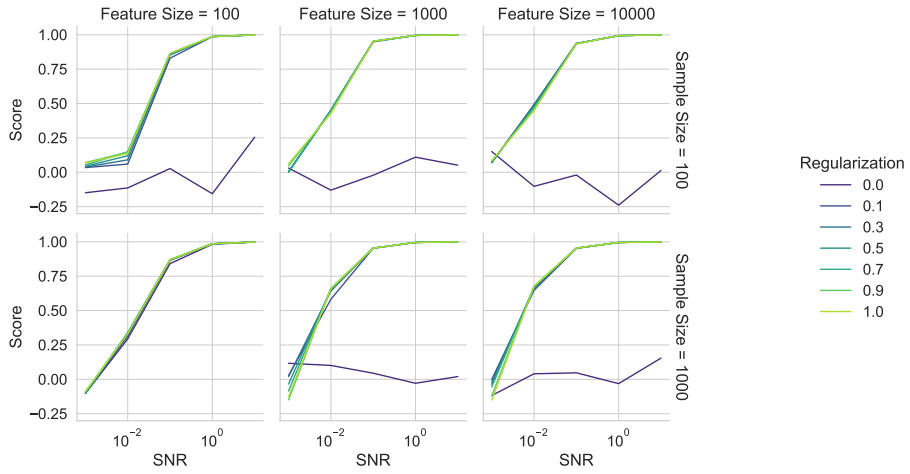


Figure IV.7: Varying signal to noise ratio with identity covariance matrices. We plot the performance of different levels of Regularized CCA from 0 (CCA) to 1 (PLS) for different sample sizes.

matrices respectively.

In figure IV.7, we can see that the PLS model outperforms all of the Ridge CCA models for all values of the signal-to-noise ratio and dimensionality, though only by a small margin. The unregeularized CCA model is much worse than even the Ridge CCA model with the smallest regularization. In this experiment the performance of PLS is directly related to the signal-to-noise ratio.

In figure IV.8, we see a totally different picture. The PLS model is now outperformed by the Ridge CCA model with the smallest regularization. While CCA is still the worst performing model, PLS is now much worse across signal-to-noise ratios and dimensions than any of the Ridge CCA models. This suggests that the PLS model is not able to recover anything like the true signal when the covariance matrices are correlated.

In this experiment it is also clear that the signal-to-noise ratio must be higher to obtain the same performance with higher dimensional data. It is interesting that performance of the Ridge CCA improves across the board with lower regularization.

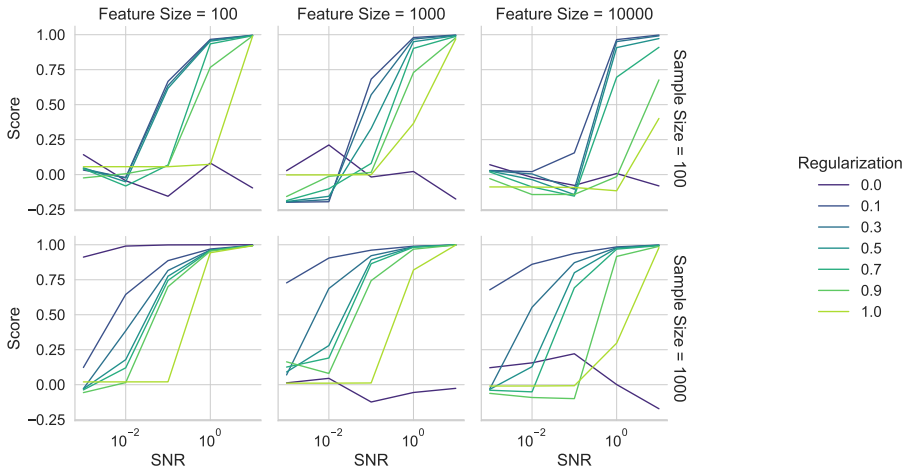


Figure IV.8: Varying signal to noise ratio with correlated covariance matrices. We plot the performance of different levels of Regularized CCA from 0 (CCA) to 1 (PLS) for different sample sizes.

8 Discussion

8.1 Revisiting the results from chapter III

In appendix A, we revisit the results from chapter III in the context of the theoretical results from this chapter. While they are not the focus of this chapter, they provide a useful comparison and point of reference for the results in this chapter.

8.2 Future Work

Given our theoretical observations in this chapter, 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:

$$P(W) = \|W\|_1 \tag{IV.34}$$

$$P(L) = \|\hat{\Sigma}U\|_1 \tag{IV.35}$$

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.

8.3 Conclusion

In this chapter, we explored the relationship between weights and loadings in CCA models from both theoretical and empirical perspectives. We unified methods for generating simulated multiview data using implicit and explicit latent variable models, providing a framework for understanding the properties of CCA and PLS models.

Through a rigorous mathematical argument, we demonstrated that loadings are invariant to columnwise transformations of the data matrix, while weights are not. This invariance property makes loadings a more reliable choice for interpreting CCA models, as the weights can be arbitrarily set by scaling the data matrix or adding linear combinations of columns.

Our experiments using simulated data provided empirical evidence supporting the theoretical findings. We showed that the recovery of true weights and loadings depends on the underlying covariance structure and the choice of regularization. The results highlighted the importance of considering the signal-to-noise ratio and dimensionality when applying CCA and PLS models to real-world datasets.

Overall, this chapter contributes to a better understanding of the behavior and interpretation of CCA models, providing valuable insights for researchers and practitioners working with multiview data. The findings emphasize the importance of considering the invariance properties of loadings and the impact of covariance structure and regularization on model performance. Future research could explore the extension of these insights to more complex data scenarios and the development of efficient algorithms for imposing sparsity on loadings in CCA models.

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
>
> Chris Ré

(I. Gemp, McWilliams, et al., 2021)

Contents

Preface

The content of this chapter is based on a series of papers (Chapman, Aguila, and Wells, 2022; Chapman, Wells, and Aguila, 2024) as well as a NeurIPS workshop paper (Chapman and Wells, 2023). 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.

1 Introduction

Classical algorithms for linear CCA methods require computing full covariance matrices and so scale quadratically with dimension, becoming intractable for many large-scale datasets of practical interest. There is therefore great interest in approximating solutions for CCA in stochastic or data-streaming settings (Arora, Cotter, et al., 2012).

In this chapter, we propose a novel approach to solving the Generalized Eigenvalue Problem (GEP) for CCA and related methods. Our method is based on a novel objective function inspired by the Eckhart–Young–Minsky inequality (Stewart and J.-G. Sun, 1990). We show that this objective has no spurious local minima and can be optimized efficiently using stochastic gradient descent. We apply this method to CCA and Partial Least Squares (PLS), and show that it can outperform existing methods in terms of convergence speed and robustness to hyperparameter settings.

2 Background: Solutions to CCA

2.1 Solving High-Dimensional Generalized Eigenvalue Problems

The GEP is often represented as $Au = \lambda Bu$, 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 $p_1 + p_2$, as A and B in these cases are $(p_1 + p_2) \times (p_1 + p_2)$.

2.2 Unified GEP formulation for CCA, Ridge CCA, PLS, and PCA

As discussed in II4, we can formulate a unified framework for CCA, ridge-regularized CCA, and PLS using a generalized eigenvalue problem (GEP). This framework involves block matrices $A, B_\alpha \in \mathbb{R}^{D \times D}$, where the diagonal blocks of A are zero, the off-diagonal blocks of B_α are zero, and the remaining blocks are defined as:

$$A^{(ij)} = \mathrm{Cov}(X^{(i)}, X^{(j)}) \text{ for } i \neq j, \tag{V.1}$$

$$B_\alpha^{(ii)} = \alpha_i I_{D^{(i)}} + (1 - \alpha_i)\mathrm{Var}(X^{(i)}), \tag{V.2}$$

where $\alpha \in [0, 1]^I$ is a vector of ridge penalty parameters. By adjusting the values of α, we can recover various subspace learning methods:

- Pure CCA: Setting $\alpha_i = 0 : \forall i$ recovers the classic CCA problem.
- Ridge Extensions: Smoothly transition to ridge-regularized CCA or PLS by selectively adjusting values within α.

- PCA Subsumption: Even PCA emerges as a special case – a single-view form of ridge-regularized PLS.

Crucially, this unification allows us to focus on solving the core CCA problem for the remainder of the chapter. The insights and solutions we develop will naturally generalize to the entire family of CCA, PLS, ridge-regularized extensions, and even PCA.

2.3 Classical Methods for Solving CCA

2.3.1 Direct Solution

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 $\mathcal{O}((p_1+p_2)^3)$ and may suffer from numerical instability.

2.3.2 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 $\mathcal{O}(p_1^3 + p_2^3)$, 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 $\mathcal{O}((2K)^3)$. Thus, the overall complexity of PCA-CCA is $\mathcal{O}(p_1^3 + p_2^3) + (2K)^3)$, 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_1 or p_2, i.e. $n < p_1$ or $n < p_2$. In this case the maximum number of principal components is $K = n$, and the complexity of PCA is $\mathcal{O}(n^3 + n^3)$ so that the overall complexity of PCA-CCA is thus $\mathcal{O}(2n^3 + (2n^3)^3)$ = $\mathcal{O}(10n^3)$. For fat data where p_1 and p_2 are larger than n, we can reasonably expect $10n^3 < p_1^3 + p_2^3$ and thus PCA-CCA is still more efficient than solving the original GEP.

We illustrate this in a simple simulation study in Figure V.1[1].

[1] This simulation was used to justify our pull request to `scikit-learn`(Pedregosa et al., 2011) implementing a PCA-PLS and PCA-CCA backend

Figure V.1: Comparison of the complexity of PCA-CCA and CCA for varying numbers of samples and features.

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.

2.3.3 Kernel CCA

Kernel CCA (KCCA) also offers computational efficiency for high-dimensional data ($p_i > n$) as its complexity scales with the number of samples n, not the number of features p_i(Akaho, 2006). It casts the CCA optimisation as a dual problem:

$$\alpha_{\text{opt}} = \underset{\alpha}{\text{argmax}}\{\alpha^{(1)} K^{(1)\,T} K^{(2)} \alpha^{(2)}\} \tag{V.3}$$

subject to:

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

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

Where $\alpha^{(i)} =$ are dual variables, $K^{(i)}$ are kernel matrices, and $K^{(i)\,T}$ are their transposes. The kernel matrices are defined as $K^{(i)} = \phi(X^{(i)})\phi(X^{(i)})^T$, where $\phi(\cdot)$ is a nonlinear mapping function. The kernel trick is used to avoid the explicit computation of the nonlinear mapping function $\phi(\cdot)$. The complexity of KCCA is

$\mathcal{O}(n^3)$, which can be much lower than the complexity of solving the original GEP directly when $p_i > n$. 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. Furthermore, when the number of samples is large, the kernel matrix can itself be too large to fit in memory.

2.4 Stochastic Algorithms for CCA

2.4.1 Classical Iterative Algorithms for PCA

In order to gain the intuition behind a number of stochastic algorithms for CCA, it's essential to understand the classical iterative algorithms for generalized eigenvalue problems, such as Sanger's rule Sanger, 1989 (the generalized Hebbian algorithm (GHA)) and Oja's rule.

The power method is a simple iterative algorithm for finding the dominant eigenvector of a matrix A. The update rule for the power method is:

$$u \leftarrow \frac{Au}{\|Au\|} \tag{V.4}$$

where u is the current estimate of the dominant eigenvector. The power method converges to the dominant eigenvector of A under mild conditions.

Sanger's rule (Sanger, 1989), also known as the GHA, is an iterative learning rule for finding the principal components of a data set. It can be written as:

$$U \leftarrow U + \eta \left(AU - U\mathsf{tril}(U^T AU)\right) \tag{V.5}$$

where $\mathsf{tril}(\cdot)$ denotes the lower triangular part of a matrix.

Oja's rule (Oja, 1982), a simplified version of Sanger's rule, can be written as:

$$U \leftarrow U + \eta \left(AU - U(U^T AU)\right) \tag{V.6}$$

In order to maintain orthogonality of the eigenvectors during the iterative updates, Oja's rule can be combined with a QR decomposition step:

$$U \leftarrow U + \eta \left(AU - U(U^T AU)\right) \tag{V.7}$$

$$U \leftarrow \mathsf{qr}(U) \tag{V.8}$$

where $\mathsf{qr}(\cdot)$ denotes the QR decomposition, which factorizes a matrix into an or-

thogonal matrix Q and an upper triangular matrix R. By using only the orthogonal matrix Q, we ensure that the eigenvectors remain orthogonal throughout the iterative process. Alternatively, other orthogonalization techniques such as Gram-Schmidt process (Schmidt, 1907; Wong, 1935) can be employed.

Both Sanger's rule and Oja's rule aim to find the principal subspace of a matrix A, which is equivalent to solving the eigenvalue problem $AU = U\Lambda$ where Λ is a diagonal matrix of eigenvalues. These algorithms can be seen as iterative methods for solving the optimization problem:

$$\max_{U} \operatorname{Tr}\left(U^T A U\right) \quad \text{subject to} \quad U^T U = I \tag{V.9}$$

GHA can also be extended to solve the generalized eigenvalue problem $AU = BU\Lambda$, where B is a positive definite matrix(Z. Chen et al., 2019). The update rule for this problem is:

$$U \leftarrow U + \eta \left(AU - BU\text{tril}(U^T A U)\right) \tag{V.10}$$

which subsumes the original GHA as a special case when $B = I$.

A number of algorithms have been proposed to approximate GEPs including PCA and PLS (Arora, Cotter, et al., 2012), and CCA specifically (K. Bhatia et al., 2018), in the 'stochastic' or 'data-streaming' setting; these can have big computational savings.

2.5 Stochastic Power Method

Arora, Mianjy, and Marinov (2016) demonstrate that PLS can be approximated by applying a stochastic Power Method. For PLS, the iterations are computed by:

$$U^{(1)}t = \mathcal{P}\text{orth}\left(U^{(1)}t - 1 + \eta_t X_t Y_t^\top U^{(2)}t - 1\right),$$
$$U^{(2)}t = \mathcal{P}\text{orth}\left(U^{(2)}t - 1 + \eta_t Y_t X_t^\top U^{(1)}t - 1\right),$$

where $\mathcal{P}_{\text{orth}}(\cdot)$ represents an orthogonal projection operator that projects a vector or matrix onto the space orthogonal to the current subspace. $\hat{U}_t^{(1)}$ and $\hat{U}_t^{(2)}$ are the current estimates of the left and right singular vectors for the two views, X_t and Y_t are the new data points at time t, and η_t is the learning rate at time t. The approach has low computational complexity, $\mathcal{O}(k(p_1 + p_2))$. However, there are two main drawbacks to this approach. First, convergence is not guaranteed. Second, the orthogonal projection step does not extend naturally to the CCA problem where the

constraint is $U^\top BU = I$ rather than $U^\top U = I$.

2.5.1 SGHA

SGHA is an algorithm for finding the top-k generalized eigenvectors of a matrix pair ((A, B)), where A is symmetric and B is symmetric positive definite. It builds upon the generalized Hebbian algorithm (GHA). Specifically, they form the constrained optimization problem for the top-k subspace as:

$$\min_U -\operatorname{Tr}\left(U^T AU\right) \quad \text{subject to} \quad U^T BU = I \tag{V.11}$$

Transforming this into an unconstrained problem using Lagrange multipliers:

$$\min_U -\operatorname{Tr}\left(U^T AU\right) + \lambda\left(U^T BU - I\right) \tag{V.12}$$

Differentiating with respect to U and λ gives the stationary points:

$$2AU - 2BU\lambda = 0 \quad \text{and} \quad U^T BU - I = 0 \tag{V.13}$$

$$\implies \lambda = U^T AU \tag{V.14}$$

Given this, the authors propose a primal dual update rule:

$$U \leftarrow U - \eta\left(AU - BU\lambda\right) \lambda \leftarrow \left(U^T AU\right) \tag{V.15}$$

where η is a learning rate.

These updates can also be combined into a single update rule:

$$U \leftarrow U - \eta\left(AU - BU\left(U^T AU\right)\right) \tag{V.16}$$

This algorithm is very simple to implement but because it is based on a heuristic primal-dual update rule rather than gradient descent, it is hard to use with more sophisticated optimizers such as Adam (Kingma and Ba, 2014).

2.5.2 γ-EigenGame

The γ-EigenGame is a stochastic algorithm for CCA inspired by the γ-EigenGame algorithm for PCA. The key idea behind the EigenGame algorithms is to view the eigenvectors as competing players trying to explain the data. In this game-theoretic

perspective, each eigenvector (player) u_i aims to maximize its own utility function:

$$\max_{u_i} \overbrace{\frac{u_i^T A u_i}{u_i^T B u_i}}^{\text{rewards}} - \overbrace{\sum_{j<i} \frac{(u_j^T A u_j)(u_i^T B u_j)^2}{(u_j^T B u_j)^2 (u_i^T B u_i)}}^{\text{penalties}} \tag{V.17}$$

The utility function consists of a reward term, $\frac{u_i^T A u_i}{u_i^T B u_i}$, which encourages the eigenvector to align with the direction of maximum variance in the data, and a penalty term, $\sum_{j<i} \frac{(u_j^T A u_j)(u_i^T B u_j)^2}{(u_j^T B u_j)^2 (u_i^T B u_i)}$, which discourages the eigenvector from aligning with the directions already captured by the previous eigenvectors.

A key advantage of the γ-EigenGame formulation is that each player i only needs to maintain orthogonalization with respect to the previous players $j < i$. This allows for a more efficient and decentralized computation of the eigenvectors.

By applying a few heuristic arguments, the γ-EigenGame algorithm derives an update rule for each eigenvector in the full batch case:

$$u_i \leftarrow (u_i^T B u_i) A u_i - (u_i^T A u_i) B u_i - \sum_{j<i} (u_i^T A y_j)[(u_i^T B u_i) B y_j - (u_i^T B y_j) B u_i] \tag{V.18}$$

where $y_j = \frac{u_j}{\sqrt{u_j^T B u_j}}$. This update rule adjusts the eigenvector u_i to maximize its utility, considering the rewards and penalties.

In the stochastic version of the γ-EigenGame algorithm, the update rule is modified to use a rolling average of the matrix B, introducing an additional hyperparameter γ that needs to be tuned. This stochastic update helps to reduce the computational complexity and memory requirements of the algorithm, making it more suitable for large-scale problems.

To the best of our knowledge, the state-of-the-art in Stochastic PLS and CCA are the subspace Generalized Hebbian Algorithm (SGHA) (Z. Chen et al., 2019) and γ-EigenGame (I. M. Gemp et al., 2020; I. Gemp, McWilliams, et al., 2021).

2.5.3 Further Benefits of Stochastic Algorithms

Stochastic algorithms not only offer computational advantages but also introduce a form of implicit regularization (S. L. Smith et al., 2021), which can be particularly beneficial in high-dimensional settings. Regularization techniques are commonly used to prevent overfitting and improve the generalization performance of models, especially when dealing with limited data or complex models.

As we have seen, in batch learning algorithms, regularization is often explicit, such as adding L1 or L2 penalty terms to the objective function. These penalty terms constrain the model's parameters and discourage overfitting. However, stochastic algorithms inherently introduce regularization through the noise in the stochastic updates, without the need for explicit penalty terms.

The intuition behind the regularizing effect of stochastic algorithms can be understood as follows:

- **Stochastic updates introduce noise**: In each iteration of a stochastic algorithm, the update is based on a random subset of the data (a mini-batch) rather than the entire dataset. This introduces noise in the updates, causing the model's parameters to fluctuate around their true values. The noise can be seen as a form of random perturbation that helps the algorithm explore the parameter space and escape from shallow local minima.
- **Noisy updates prevent overfitting**: The noise in the stochastic updates acts as a form of regularization that prevents the model from fitting too closely to the training data. By introducing this randomness, stochastic algorithms reduce the risk of overfitting, especially when the model has a large number of parameters relative to the size of the training data. The noisy updates effectively smooth out the loss function, making it harder for the model to memorize individual training examples.
- **Averaging effect**: As the stochastic algorithm progresses, the noisy updates tend to cancel out each other over time, leading to an averaging effect. This averaging helps the model converge towards a solution that generalizes well to unseen data, rather than getting stuck in a suboptimal solution that overfits the training data. Intuitively, the averaging effect can be seen as a form of ensembling, where multiple noisy models are combined to produce a more robust and generalizable solution.
- **Implicit bias towards simpler models**: The noise in the stochastic updates implicitly biases the algorithm towards simpler models that are less prone to overfitting. This is because simpler models are more stable under random perturbations, while complex models that overfit the data are more sensitive to noise. As a result, stochastic algorithms tend to favor models with smoother decision boundaries and better generalization performance, even without explicit regularization terms in the objective function.

The regularizing effect of stochastic algorithms has been extensively studied in the context of deep learning (C. Zhang et al., 2021; Chaudhari and Soatto, 2018). These studies have shown that the implicit regularization introduced by stochastic updates can lead to improved generalization performance and robustness to noise, especially in high-dimensional settings where the number of parameters is large compared to the number of training examples.

2.6 Limitations of Existing Stochastic CCA Algorithms

While stochastic algorithms like SGHA and γ-EigenGame have made significant progress in addressing the computational challenges of CCA in high-dimensional settings, they still have some limitations that motivate the development of novel objectives and algorithms.

One limitation of SGHA is that it relies on a heuristic primal-dual update rule rather than a principled optimization framework. This makes it difficult to integrate with more sophisticated optimizers like Adam (Kingma and Ba, 2014), which have been shown to improve convergence speed and stability in many machine learning applications.

On the other hand, γ-EigenGame requires the tuning of a hyperparameter γ in the stochastic setting. This hyperparameter controls the trade-off between the computational efficiency and the accuracy of the stochastic updates, and its optimal value may vary depending on the problem and the data.

To address the limitations of existing stochastic CCA algorithms and provide a more principled and robust approach, we propose a novel formulation of the CCA problem based on the Eckhart–Young–Minsky inequality (Stewart and J.-G. Sun, 1990). Our formulation leads to a new objective function that characterizes the top-K subspace of Generalized Eigenvalue Problems (GEPs), including CCA as a special case.

3 Methods: Novel Objectives and Algorithms

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 B1.

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 $U \in \mathbb{R}^{D\times K}$:*

$$\mathcal{L}_{\textit{EY-GEP}}(U) := \operatorname{trace}\left(-2\,U^\top AU + \left(U^\top BU\right)\left(U^\top BU\right)\right) \tag{V.19}$$

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

The objective in Equation equation V.19 has an intuitive interpretation. The first term, $-2\operatorname{trace}(U^\top AU)$, acts as a reward for high covariance between the views, encouraging the learned subspace to capture the directions of maximum correlation. The second term, $\operatorname{trace}((U^\top BU)(U^\top BU))$, serves as a penalty for high variance within each view and promotes orthogonality between the learned components. By minimizing this objective, we aim to find a subspace that maximizes the correlation between views while ensuring the learned components are distinct and informative.

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. *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 Ge, Jin, and Zheng (2017) and Ge, Huang, et al. (2015); we state an informal version here and give full details in B2.

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.

3.1 Corresponding Objectives for the CCA family

For the case of linear CCA we have $U^\top AU = \sum_{i\neq j}\operatorname{Cov}(Z^{(i)}, Z^{(j)})$, $U^\top BU = \sum_i \operatorname{Var}(Z^{(i)})$.

To generalize this to multiview CCA, we define the analogous matrices of total between-view covariance and total within-view variance

$$C(\theta) = \sum_{i \neq j} \mathrm{Cov}(Z^{(i)}, Z^{(j)}), \quad V(\theta) = \sum_i \mathrm{Var}(Z^{(i)}) \tag{V.20}$$

While for ridge CCA we can add a ridge penalty to the within-view variances:

$$V_\alpha(\theta) = \sum_i \alpha_i U^{(i)\,T} U^{(i)} + (1 - \alpha_i)\mathrm{Var}(Z^{(i)}) \tag{V.21}$$

This leads to the 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*

$$\mathcal{L}_{EY}(\theta) = -2 \operatorname{trace} C(\theta) + \|V_\alpha(\theta)\|_F^2 \tag{V.22}$$

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

$$\hat{C}(\theta)[\mathbf{Z}] = \sum_{i \neq j} \widehat{\mathrm{Cov}}(\mathbf{Z}^{(i)}, \mathbf{Z}^{(j)}), \quad \hat{V}(\theta)[\mathbf{Z}] = \sum_i \widehat{\mathrm{Var}}(\mathbf{Z}^{(i)}) \tag{V.23}$$

In the linear case we can construct $\hat{V}_\alpha(\theta)[\mathbf{Z}]$ analogously by plugging sample covariances into Equation (V.21). Then if $\mathbf{Z}, \mathbf{Z}'$ are two independent batches of transformed variables, the batch loss

$$\hat{\mathcal{L}}_{\mathsf{EY}}[\mathbf{Z}, \mathbf{Z}'] := -2 \operatorname{trace} \hat{C}[\mathbf{Z}] + \langle \hat{V}_\alpha[\mathbf{Z}], \hat{V}_\alpha[\mathbf{Z}'] \rangle_F \tag{V.24}$$

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 θ.

Simple algorithms: We first define a general algorithm using these estimates in Algorithm 1. In the next section we apply this algorithm to multiview stochastic CCA (**CCA-EY**) and PLS (**PLS-EY**).

3.2 Applications to multiview stochastic CCA and PLS

Lemma 3.1 (Objective recovers GEP formulation of linear multiview CCA)**.** *When the* $f^{(i)}$ *are linear, the population loss from Equation* (V.22) *recovers MCCA.*

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 θ.
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.24)
  Obtain gradients by back-propagation and step with your favourite optimizer.
end for

Proof. By construction, for linear MCCA we have $C = U^\top A U$, $V_\alpha = U^\top B_\alpha U$, where (A, B_α) define the GEP for MCCA introduced in **??**. So $\mathcal{L}_{\mathsf{EY}}(U) = \mathcal{L}_{\mathsf{EY\text{-}GEP}}(U)$ 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 Section 3.

4 Experiments and Results

4.1 Comparison to Scipy

In a first simple experiment, we compare our method to solving the CCA Generalized Eigenvalue Problem using the `scipy` implementation of `eigh` (Virtanen et al., 2020). We use a sample size of 10,000 and vary the number of features in each view from 10 to 5,000. We solve for the top-5 CCA subspace and repeat each experiment 5 times. We compare the time taken to solve the GEP using `eigh` to the time taken to train our method until the frobenius norm of the change in successive weights is less than 10^{-5}.

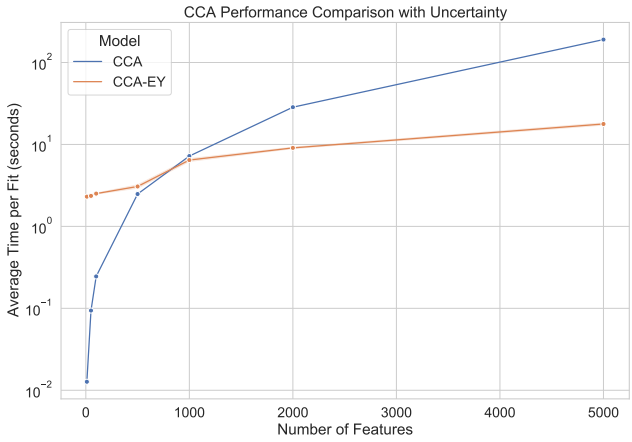


Figure V.2: Comparison of the time taken to solve CCA using `eigh` and our CCA-EY method.

4.1.1 Observations

Figure V.2 shows the results of this experiment. Up until 1,000 features, our method is slower than `eigh` but after this point, our method is significantly faster. Beyond this point, the time taken to solve the GEP using `eigh` scales quadratically with the number of features, while our method scales linearly. This is because our method scales linearly with both the number of samples and the number of features, while `eigh` scales quadratically with the number of features[2].

4.2 Stochastic CCA

In our second experiment, we aim to demonstrate that our proposed CCA-EY method not only matches but potentially surpasses the performance of established baselines γ-EigenGame and SGHA in terms of convergence speed and robustness to hyperparameter settings. Our experimental setup largely follows the framework established by Z. Meng, Chakraborty, and Singh (2021) and I. Gemp, C. Chen, and McWilliams (2022). A key distinction in our approach, however, is the decision to not perform PCA on the data prior to applying CCA methods. This choice retains the full complexity of the datasets, providing a more rigorous evaluation of each algorithm's

[2]When the number of features is greater than the number of samples as in ridge CCA and PLS, we can use the PCA or Kernel trikcs to scale CCA quadratically with the minimum of the number of features and samples, though this still requires a somewhat expensive PCA

ability to handle high-dimensional data efficiently and accurately.

One of the central goals of this comparison is to illustrate that CCA-EY can achieve faster convergence with less hyperparameter tuning, an essential attribute for practical applications. To facilitate a fair and direct comparison with the baseline methods, we employ Stochastic Gradient Descent (SGD) as the optimization technique for all algorithms. It is worth noting that while SGD provides a baseline for performance assessment, the potential of our CCA-EY method could be further unleashed by utilizing more advanced optimization techniques such as momentum-based optimizers like Adam or Nesterov acceleration. These advanced methods are known for their ability to accelerate convergence and navigate the optimization landscape more effectively, suggesting that our method might yield even better performance under such enhanced optimization schemes.

We train models to optimize CCA on the MediaMill and Split-CIFAR-10 datasets for a single epoch, using mini-batch sizes ranging from 5 to 100. These sizes were selected to test the scalability and efficiency of our method under varied computational loads. The Proportion of Correlation Captured (PCC) metric, defined as PCC $= (\sum_{i=k}^{K} \rho_k)/(\sum_{k=1}^{K} \rho_k^*)$, serves as our evaluation criterion. Here, ρ_k represents the correlations of the estimated representations $Z^{(i)} = X^{(i)}\hat{U}^{(i)}$ with one another on the test set, while ρ_k^* denotes the canonical correlations computed from the full batch covariance matrices. In other words, using our earlier notation, $\rho_k = \mathrm{MCCA}_K(\hat{Z}^{(1)}, \hat{Z}^{(2)})$ and $\rho_k^* = \mathrm{MCCA}_K(X^{(1)}, X^{(2)})$.

Despite ρ_k^* not being the 'true' correlations, their computation from a large sample size renders them a reliable benchmark. PCC is an efficient metric for tracking algorithmic performance over time, minimizing computational overhead(Z. Meng, Chakraborty, and Singh, 2021; I. Gemp, C. Chen, and McWilliams, 2022; Ma, Lu, and Foster, 2015; Ge, Jin, Netrapalli, et al., 2016).

4.2.1 Data

The MediaMill dataset (Snoek et al., 2005) comprises paired features of videos and corresponding commentary, with the objective of learning joint representations that capture their correlation. This representation could potentially enable prediction of commentary from video, or vice versa. The dataset includes 25,800 test images, with 120 and 101 features respectively.

The Split-CIFAR dataset (Z. Meng, Chakraborty, and Singh, 2021) consists of 50,000 training and 10,000 test RGB images, each split in half with 32x16x3 features. The aim is to learn joint representations of the two halves that reveal

correlations, expected to be high within the same class and low across different classes. These datasets are chosen for their diverse nature and complexity, providing a comprehensive test bed for our method.

4.2.2 Parameters

For each method, we searched over the hyperparameter (see Table 4.1) 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
γ^3	0.01,0.1,1,10

Table 4.1: Hyperparameter ranges explored for CCA methods.

4.2.3 Observations

In machine learning, a learning curve represents a graph showing the model's learning progress over time in terms of experience or iterations. For the MediaMill dataset, Figure V.3 compares the algorithms' learning curves for various mini-batch sizes, showing CCA-EY's consistent outperformance. Figure V.4 further examines the learning curves for batch sizes 5 and 100, illustrating CCA-EY's superior performance over time.

For the CIFAR dataset, Figure V.5 shows the performance comparison across batch sizes, while Figure V.6 details the learning curves, highlighting the underperformance of γ-EigenGame, especially for smaller batch sizes.

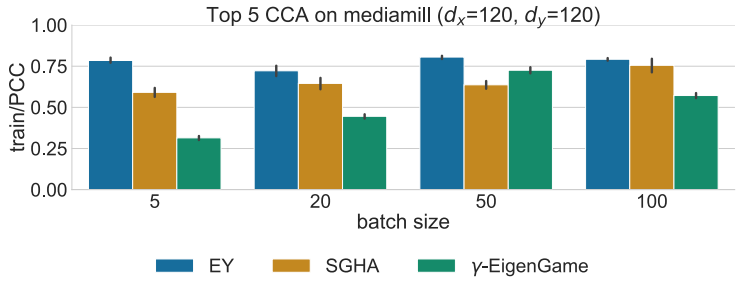


Figure V.3: Stochastic CCA on MediaMill using PCC: Performance across varying mini-batch sizes. Shaded regions signify $\pm$ one standard deviation around the mean of 5 runs.

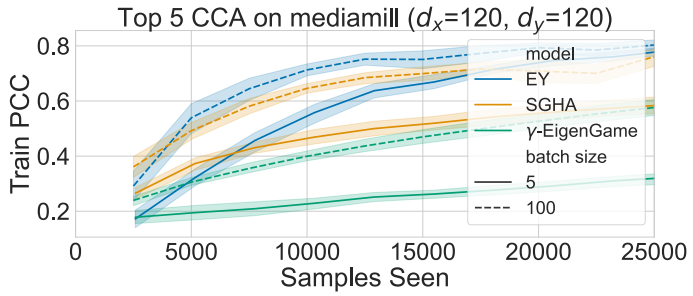


Figure V.4: Stochastic CCA on MediaMill: Training progress over a single epoch for mini-batch sizes 5, 100.

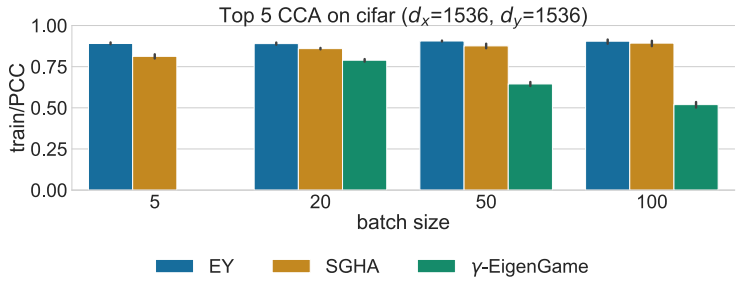


Figure V.5: Stochastic CCA on CIFAR using PCC: Performance across varying mini-batch sizes. Shaded regions signify $\pm$ one standard deviation around the mean of 5 runs.

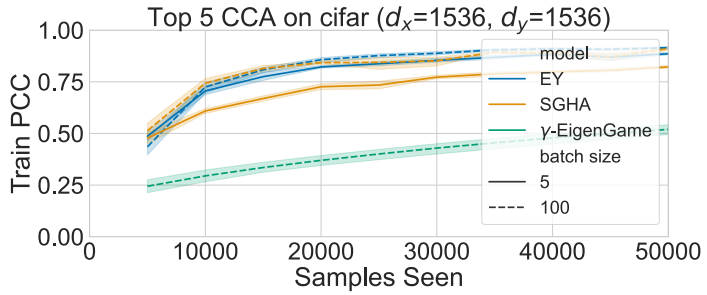


Figure V.6: Stochastic CCA on CIFAR: Training progress over a single epoch for mini-batch sizes 5, 100.

4.3 Stochastic PLS UK Biobank

In this section, we aim to demonstrate the exceptional scalability and efficiency of our Stochastic PLS method, PLS-EY, in handling extremely high-dimensional imaging genetics data. We employ imaging genetics data from the UK Biobank (Sudlow et al., 2015) as our test bed, given its comprehensive and complex nature. The UK Biobank dataset presents a unique challenge due to the sheer scale of its genetic data, requiring sophisticated regularization strategies.

PLS is particularly suited for imaging-genetics studies due to its capability to handle high dimensionality and reveal novel phenotypes as well as genetic mechanisms underlying diseases and brain morphometry. Historically, imaging genetics analyses have been constrained to smaller datasets due to computational limitations (Lorenzi, Altmann, et al., 2018; Taquet et al., 2021; Le Floch et al., 2012). Moreover, the few studies that have attempted to analyze data of comparable scale to the UK Biobank have typically resorted to partitioning the data into smaller clusters, thereby limiting the scope of their analysis (Lorenzi, Gutman, et al., 2017; Altmann et al., 2023).

Our experiment with PLS-EY, conducted on a subset of the UK Biobank dataset consisting of brain imaging data (82 regional volumes) and genetic data (582,565 variants) for 33,333 subjects, is designed to overcome these limitations. A particular computational challenge we address is maintaining orthogonality between the weight vectors u_k in the PLS model, which is crucial for the method's effectiveness. We run PLS-EY with a mini-batch size of 500 and train the GEP-EY PLS analysis for 100 epochs using a learning rate of 0.0001. This approach allows us to not only manage the high-dimensional nature of the data but also to preserve the integrity and interpretability of the analysis. To our knowledge, this represents the largest-scale

PLS analysis of biomedical data to-date, showcasing the potential of our method to facilitate discoveries in extremely large datasets.

4.3.1 Data

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.

4.3.2 Observations

We observed strong validation correlations between all 10 corresponding pairs of representations $Z_k^{(1)}$ and $Z_k^{(2)}$ in the PLS model, with weak cross-correlations between $Z_k^{(1)}$ and $Z_i^{(2)}$ for $i \neq k$. This indicates that our model learned a coherent and orthogonal subspace, as shown in Figure V.7. Furthermore, the PLS representations Z were significantly associated with genetic risk measures for several disorders, suggesting that the learned PLS subspace encodes relevant information for genetic disease risk, a critical insight for biomedical research (Figure V.8). These results demonstrate the scalability of our method to extremely high-dimensional data, and its ability to learn interpretable representations.

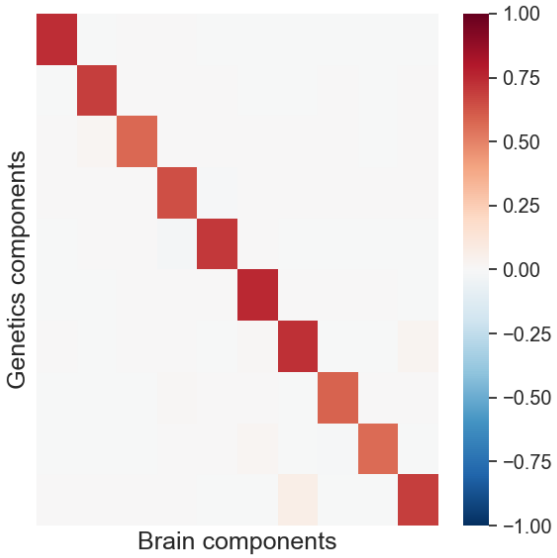


Figure V.7: Pearson correlations among PLS latent variables Z_k derived from UK Biobank data.

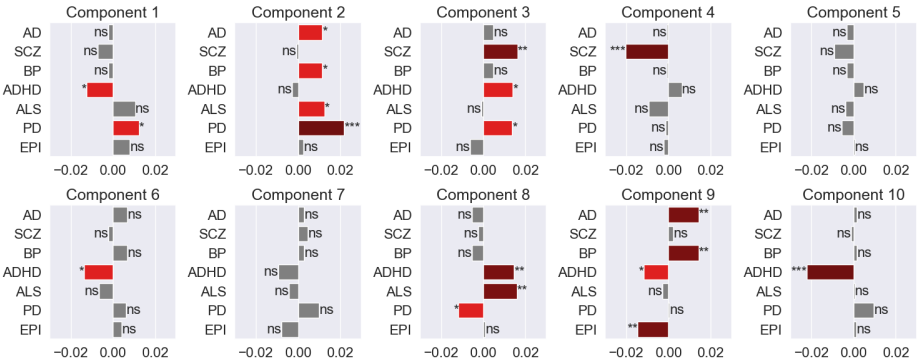


Figure V.8: Correlation between PLS brain representations Z and genetic risk scores for various disorders. AD=Alzheimer's disease, SCZ=Schizophrenia, BP=Bipolar, ADHD=Attention deficit hyperactivity disorder, ALS=Amyotrophic lateral sclerosis, PD=Parkinson's disease, EPI=Epilepsy. ns : $0.05 < p \leq 1, * : 0.01 < p \leq 0.05, ** : 0.001 < p \leq 0.01, *** : 0.0001 < p \leq 0.001$.

5 Discussion

5.1 Limitations

This chapter presents a comprehensive exploration and development of novel algorithms for Canonical Correlation Analysis (CCA) and Partial Least Squares (PLS), focusing on scalability and efficiency in high-dimensional and large-scale datasets. Our approach introduces the Eckhart-Young (EY) inspired objectives for Generalized Eigenvalue Problems (GEPs) and their application in stochastic or data-streaming settings, paving the way for more efficient and scalable solutions to classical subspace learning problems.

Our proposed CCA-EY and PLS-EY methods demonstrate significant advancements over traditional approaches in handling the computational complexity and scalability issues inherent in high-dimensional data. By reformulating the CCA and PLS objectives, we provide a path to efficiently analyze large datasets, which was previously infeasible due to computational limitations. The empirical evaluation on diverse datasets, including MediaMill, Split-CIFAR-10, and the UK Biobank, not only validates the effectiveness of our methods but also highlights their superiority in convergence speed and robustness to hyperparameter tuning.

The results from the MediaMill and Split-CIFAR-10 datasets underscore the potential of CCA-EY in achieving faster convergence with minimal hyperparameter tuning, a crucial factor for practical applications. This advantage is particularly pronounced when comparing our method to established baselines like γ-EigenGame and SGHA. Additionally, the application of our methods to the UK Biobank dataset represents a breakthrough in the analysis of imaging genetics data, showcasing the capability of PLS-EY to manage extraordinarily high-dimensional data while extracting meaningful and interpretable representations.

Furthermore, our methods' ability to capture relevant information for genetic disease risk, as evidenced in the UK Biobank study, opens new avenues for biomedical research. The significant associations between the PLS representations and genetic risk measures for various disorders provide valuable insights into the genetic mechanisms underlying diseases and brain morphometry.

5.2 Future Work

5.3 Proximal Gradient Descent for Regularized GEPs

Our future initiatives will enhance CCA, PCA and PLS by incorporating proximal gradient descent for efficient handling of complex regularization terms. This methodology is ideally suited for scenarios where the loss function comprises a smooth, differentiable component plus a non-smooth regularization term. The proximal gradient technique utilizes a gradient step followed by a proximal step, enabling effective management of non-smooth penalties such as L1-norm or Total Variation (TV), which are instrumental in enforcing sparsity and structural constraints.

5.3.1 Objective Formulation with Regularization

We aim to modify the CCA framework by integrating specific regularization terms directly into the loss function of the Generalized Eigenvalue Problem (GEP). The revised loss function, denoted as $\mathcal{L}_{\text{Proximal CCA-EY}}(\mathbf{U}_1, \mathbf{U}_2)$, will incorporate the regularization terms $R_1(\mathbf{U}_1)$ and $R_2(\mathbf{U}_2)$, enabling the optimization of CCA with additional constraints. The objective function for Proximal CCA-EY will be defined as:

$$\mathcal{L}_{\text{Proximal CCA-EY}}(\mathbf{U}_1, \mathbf{U}_2) = \mathcal{L}_{\text{EY}}(\mathbf{U}_1, \mathbf{U}_2) + \lambda_1 R_1(\mathbf{U}_1) + \lambda_2 R_2(\mathbf{U}_2),$$

where $\mathcal{L}^{\text{Proximal CCA-EY}}$ represents the Proximal CCA-EY loss function, λ_1 and λ_2 are regularization parameters, and $R_1(\mathbf{U}_1)$ and $R_2(\mathbf{U}_2)$ are the regularization terms for each view.

5.3.2 Proximal Gradient Descent Mechanism

The proximal gradient updates for this augmented CCA formulation are specified as:

$$\mathbf{U}1^{(t+1)} = \text{prox}\alpha\lambda_1 R_1(\mathbf{U}_1^{(t)} - \alpha\nabla\mathbf{U}_1\mathcal{L}^{\text{EY}}(\mathbf{U}_1^{(t)}, \mathbf{U}_2^{(t)})),\ \mathbf{U}2^{(t+1)} = \text{prox}\alpha\lambda_2 R_2(\mathbf{U}_2^{(t)} - \alpha\nabla\mathbf{U}_2\mathcal{L}^{\text{EY}}(\mathbf{U}_1^{(t)}, \mathbf{U}_2^{(t)})), \tag{V.25}$$

where $\text{prox}_{\alpha\lambda_i R_i}(\mathbf{v})$ denotes the proximal operator for the regularization term R_i with parameter λ_i, and α represents the learning rate. Note that the gradients $\nabla\mathbf{U}_1\mathcal{L}^{\text{EY}}(\mathbf{U}_1^{(t)}, \mathbf{U}_2^{(t)})$ and $\nabla\mathbf{U}_2\mathcal{L}^{\text{EY}}(\mathbf{U}_1^{(t)}, \mathbf{U}_2^{(t)})$ are computed only for the smooth part of the loss function, i.e., $\mathcal{L}^{\text{EY}}(\mathbf{U}_1, \mathbf{U}_2)$, and do not include the regularization terms.

The proximal operator is defined as solving:

$$\text{prox}\alpha\lambda_i R_i(\mathbf{v}) = \arg\min \mathbf{u} \left(R_i(\mathbf{u}) + \frac{1}{2\alpha} |\mathbf{u} - \mathbf{v}|_2^2 \right).$$

This update effectively balances the influence of the gradient of the smooth loss function and the geometry imposed by the regularization, making the approach robust to the inclusion of complex constraints in the optimization of CCA.

5.3.3 Efficiency and Applicability

The proximal gradient descent method excels in large-scale optimization challenges where traditional techniques struggle due to the presence of non-smooth terms. By separating the optimization of the smooth component from the non-smooth regularization, proximal steps can be efficiently computed, particularly when R_i allows a straightforward proximal formulation.

5.4 Conclusion

In summary, this chapter contributes to the fields of machine learning and multiview data analysis by introducing scalable and efficient solutions for CCA and PLS, applicable in a variety of domains, including but not limited to neuroimaging and genetics. Our work not only addresses significant computational challenges but also lays the groundwork for future research and practical applications in analyzing large-scale, high-dimensional datasets.

Chapter VI

Deep CCA and Self-Supervised Learning: Non-Linear Functions

Contents

Preface

This chapter is based on work presented in Chapman and Wells (2023) and Chapman, Wells, and Aguila (2024).

1 Introduction

Deep CCA (Andrew et al., 2013) secured a runner-up position for the test-of-time award at ICML 2023 (ICML, 2023). However, its direct application has been limited in large datasets due to biased gradients in the stochastic minibatch setting. There have since been proposals to scale-up Deep CCA in the stochastic case with adaptive whitening (W. Wang, Arora, Livescu, and Srebro, 2015) and regularization (Chang, Xiang, and T. M. Hospedales, 2018), but these techniques are highly sensitive to hyperparameter tuning.

Self-Supervised Learning (SSL) methods have reached the state-of-the-art in tasks such as image classification (Balestriero, Ibrahim, et al., 2023), learning representations without labels that can be used to classify images using a linear probe in the zero-shot setting. A family of SSL methods that are closely aligned with Canonical Correlation Analysis (CCA) has garnered particular interest. This family notably includes Barlow Twins (Zbontar et al., 2021), VICReg (Bardes, Ponce, and LeCun, 2021), and W-MSE (Ermolov et al., 2021) and they aim to transform a pair of data views into similar representations, similar to the objective of CCA. Similarly, some generative approaches to SSL(Sansone and Manhaeve, 2022) bear a striking resemblance to Probabilistic CCA(Bach and Jordan, 2005). These connections have started to be explored in Balestriero and LeCun (2022).

In this chapter, we propose a novel formulation of Deep CCA that is unbiased in the stochastic setting and scales to large datasets. We also propose a novel SSL method, SSL-EY, that is competitive with existing methods on CIFAR-10 and CIFAR-100. We highlight the connections between our work and existing SSL methods, and show that our method is more robust to hyperparameter tuning.

2 Background: Deep Representation Learning

2.1 Deep Learning

Deep learning is a subfield of machine learning that uses functions parameterised by neural networks. Deep learning has been applied to a wide range of domains, including computer vision, speech recognition, natural language processing, and bioinformatics, where they have produced state-of-the-art results on many tasks. Neural networks are usually composed of many linear layers followed by nonlinear activation functions such as the rectified linear unit (ReLU). The ReLU activation function is defined as $\mathrm{ReLU}(x) = \max(0, x)$. The ReLU activation function is piecewise linear, and so the composition of ReLU activations with linear functions is a piecewise linear function. It has been shown that neural networks with ReLU activations can approximate any continuous function on a compact set to arbitrary accuracy (Perekrestenko et al., 2018), and so are universal function approximators. This flexibility, combined with increasingly large datasets, allows neural networks to learn complex functions from data. Owing to the size of the models and datasets, neural networks are usually trained using the backpropagation algorithm and stochastic gradient descent (SGD) (Amari, 1993).

2.2 DCCA and Deep Multiview CCA

Thus far, our focus in this thesis has been on linear CCA. However, in dealing with high-dimensional and complex data structures commonly found in modern applications, nonlinear extensions of CCA become essential. The objective of DCCA and DMCCA is to learn nonlinear representations of data that are linearly correlated across different views. Recall the definition of MCCA_K in Equation equation II.40. We define the goal of DCCA and DMCCA using this notation:

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

which is the norm of the vector of the top K canonical correlations between the representations $Z^{(i)}$ of the different views.

The key difference between MCCA and its deep learning counterparts, DCCA and DMCCA, lies in the nature of the input data. In MCCA, we work with fixed, predefined feature representations $X^{(i)}$ for each view and aim to find linear transformations that maximize the correlation between these fixed representations. In contrast,

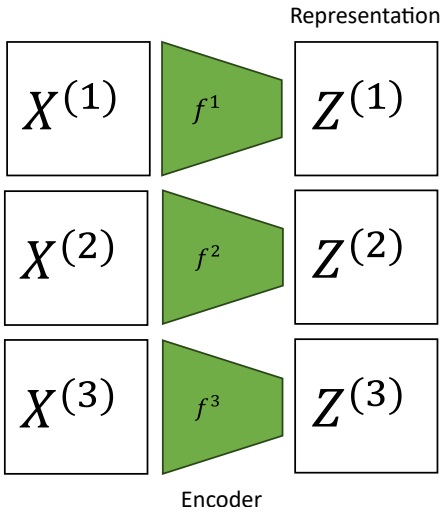


Figure VI.1: Schematic of the DCCA approach highlighting the nonlinear transformation of data into correlated views.

DCCA and DMCCA learn new, nonlinear representations $Z^{(i)} = f^{(i)}(X^{(i)}; \theta^{(i)})$ using neural networks $f^{(i)}$ parameterized by $\theta^{(i)}$ for each view $i \in [I]$. The objective is to learn representations that have high MCCA, i.e., representations that are maximally correlated across views in a linear sense.

The power of DCCA and DMCCA lies in their ability to learn flexible, nonlinear transformations of the input data that are tailored to the task of maximizing cross-view correlation. By learning these representations end-to-end, DCCA and DMCCA can potentially capture more complex and subtle relationships between views that may be difficult or impossible to capture with fixed, linear transformations.

Figure VI.1 illustrates the conceptual framework of DCCA, where data from different views are transformed through neural networks to achieve correlated representations.

The full-batch approach of DCCA, formulated by Andrew et al. (2013), seeks to maximize the correlation between these different views. The objective, operationalized as a loss function, is defined by the trace of matrix T:

$$T = \left(\text{cov}(Z^{(1)})\right)^{-\frac{1}{2}} Z^{(1)\top} Z^{(2)} \left(\text{cov}(Z^{(2)})\right)^{-\frac{1}{2}} \tag{VI.2}$$

$$\mathcal{L}_{\text{Rayleigh}} = -\operatorname{Tr}(T) \tag{VI.3}$$

This approach, while theoretically sound, faces scalability issues with large datasets. DCCA-STOL, proposed by W. Wang, Arora, Livescu, and Bilmes (2015), adapts this objective to large mini-batches but suffers from biased gradients due to the matrix inversions in Equation equation VI.2. This necessitates batch sizes larger than the representation size, limiting its practical application.

Extensions such as DMCCA (Somandepalli et al., 2019) and DGCCA (Benton et al., 2017) are multiview extensions of DCCA that work by directly differentiating the sum of the top K generalized eigenvalues of the mini-batch covariance matrices. Specifically, their loss function is given by:

$$T = \left(\hat{V}(\theta)^{-\frac{1}{2}}\hat{C}(\theta)\hat{V}(\theta)^{-\frac{1}{2}}\right) \tag{VI.4}$$

$$\mathcal{L}_{\text{MCCA}} = -\operatorname{Tr}(T) \tag{VI.5}$$

where $\hat{C}(\theta)$ and $\hat{V}(\theta)$ are the mini-batch estimates of the between-view and within-view covariance matrices, respectively, as defined in Equation equation V.20.

However, these methods suffer from the same fundamental issue as DCCA-STOL, namely the biased gradients resulting from the eigendecomposition of small mini-batch covariance matrices. As a result, DMCCA and DGCCA do not effectively mitigate the scalability issues of DCCA and still require large batch sizes to work well in practice.

Adaptive whitening methods (W. Wang, Arora, Livescu, and Srebro, 2015; Chang, Xiang, and T. M. Hospedales, 2018) offer another solution by reducing the bias in the DCCA objective. However, as noted in DCCA-NOI (W. Wang, Arora, Livescu, and Bilmes, 2015), these methods introduce a time constant that complicates analysis and requires extensive tuning:

$$\mathcal{L}_{\text{NOI}} = |\tilde{\Sigma}_{11}^{-\frac{1}{2}}Z^{(1)} - \tilde{\Sigma}_{22}^{-\frac{1}{2}}Z^{(2)}|_F^2 \tag{VI.6}$$

where $\tilde{\Sigma}_{11}$ and $\tilde{\Sigma}_{22}$ are estimates of the covariance matrices of $Z^{(1)}$ and $Z^{(2)}$, respectively.

These limitations highlight the need for more scalable and efficient nonlinear CCA methods that can handle large datasets without compromising on representation quality or requiring extensive hyperparameter tuning.

2.3 Self-Supervised Learning and Joint Embedding

Self-Supervised Learning (SSL) has emerged as a crucial approach in deep learning, especially for tasks with limited labeled data. A fundamental strategy in SSL, particularly in non-contrastive SSL, involves creating joint embeddings of augmented images. This process entails generating two distinct views of the same image, denoted as X_1 and X_2, using various augmentation techniques. The primary aim is to align their representations, $Z^{(1)}$ and $Z^{(2)}$, in a shared embedding space. This alignment leverages the inherent patterns within the data to develop feature representations absent explicit labels. A significant challenge in this methodology is averting the collapse of representations, where models produce constant features irrespective of input variability.

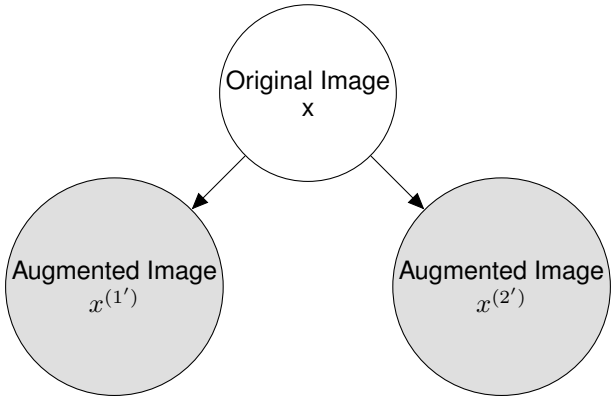


Figure VI.2: ***Joint Embedding Data Generation Process:*** The original image x is augmented to produce two views $x^{(1')}$ and $x^{(2')}$.

In SSL, augmented data generation serves to create multiple perspectives of the same underlying content, such as cropping or rotating an image. The objective is to learn representations that are invariant to these augmentations, thereby capturing the fundamental structure of the data. Canonical Correlation Analysis (CCA) emerges as a fitting tool for this task, especially given that augmentations represent redundant rather than complementary information, offering different perspectives of the same underlying data.

2.3.1 Encoder-Projector Model in SSL

SSL methods like Barlow Twins and VICReg employ an encoder-projector model, as shown in Figure VI.3. In this model, input data is transformed by an encoder

g into representations, which are further processed by a projector h into higher-dimensional embeddings. These embeddings are integral to training, with the representations being critical for downstream tasks. The encoder is typically a neural network suited to the domain, while the projector is often a simpler multi-layer perceptron.

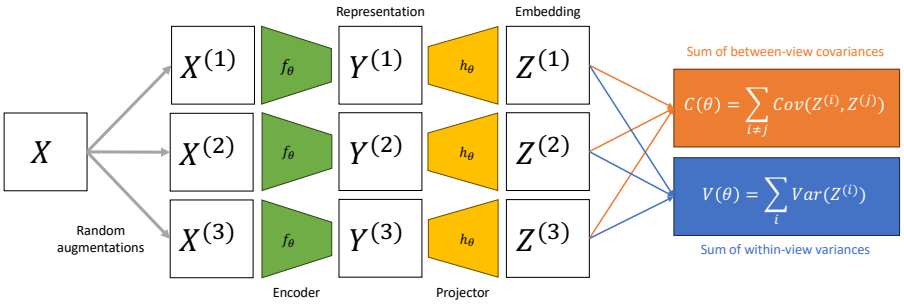


Figure VI.3: Schematic of the encoder-projector setup in SSL.

The essence of joint embedding in SSL is that similar inputs, X and its augmented version X', should yield similar embeddings, Z and Z'. The encoder and projector are optimized to minimize the distance between Z and Z', reflecting the similarity of the inputs.

2.3.2 CCA-based SSL Methods: Barlow Twins and VICReg

Barlow Twins and VICReg are two pivotal methods in SSL that build upon canonical correlation principles to generate robust representations from augmented views. Both methods aim to align representations of two augmented views while ensuring distinct yet correlated representations.

Barlow Twins employs a redundancy reduction objective to ensure similarity between representations of the same augmented views and to decorrelate representations within each view. Its loss function is expressed as:

$$\mathcal{L}_{\text{BT}} = \overbrace{\gamma \mathbb{E} \|Z^{(1)} - Z^{(2)}\|^2}^{\text{Invariance}} + \overbrace{\beta \sum_{\substack{k,l=1 \\ k \neq l}}^{K} \text{Cov}(\hat{Z}_k^{(i)}, \hat{Z}_l^{(i)})^2}^{\text{Redundancy Reduction}}, \tag{VI.7}$$

where $\hat{Z}^{(i)}$ denotes the batch-normalized versions of the representations, with γ and β as hyperparameters controlling the similarity and decorrelation terms, respectively.

VICReg, in contrast, introduces a variance term and omits batch normalization, focusing on variance-invariance-covariance regularization. The VICReg loss is defined as:

$$\mathcal{L}_{\text{VR}} = \overbrace{\gamma \mathbb{E}\|Z^{(1)} - Z^{(2)}\|^2}^{\text{Invariance}} + \left[\overbrace{\sum_{i\in\{1,2\}} \alpha \sum_{k=1}^{K} \left(1 - \sqrt{\text{Var}(Z_k^{(i)})}\right)_+}^{\text{Variance}} + \overbrace{\beta \sum_{\substack{k,l=1 \\ k\neq l}}^{K} \text{Cov}(Z_k^{(i)}, Z_l^{(i)})^2}^{\text{Covariance}} \right], \tag{VI.8}$$

with α, β, and γ as tuning parameters balancing the influence of variance, invariance, and covariance regularization.

These approaches, grounded in canonical correlation principles, offer foundational baselines for our experiments in SSL.

3 Methods: Novel Objectives and Algorithms

Recall the definition of our family of objectives from definition **??**. We will now consider non-linear transformations of the data $Z^{(i)} = f^{(i)}(X^{(i)}; \theta^{(i)})$, and show that our objectives are well-suited to this setting.

3.1 Applications to (multi-view) Deep CCA

We first show that our objective recovers Deep Multi-view CCA at any local optimum, assuming a final linear layer in each neural network.

Lemma 3.1. *[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*

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

where $\hat{Z} = f_{\hat{\theta}}(X)$. Therefore, $\hat{\theta}$ is also a local optimum of objectives from Andrew et al. (2013) and Somandepalli et al. (2019) as defined in Equation (VI.1).

Proof sketch: see Section 1 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. □

This result shows that our objective, which we call **DCCA-EY**, is a valid generalization of Deep CCA and can be used to learn correlated non-linear representations.

3.2 Application to Self-Supervised Learning (SSL)

We can directly apply Algorithm 1 to the SSL setting by treating the two augmented views as different data views. We will refer to this method as **SSL-EY**.

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 input, it is intuitive that tying the weights is a sensible procedure and may act as a regularizer[1].

Moreover, our loss function bears some resemblance to those of Barlow Twins and VICReg. Recall that our objective is:

$$\mathcal{L}_{\mathsf{EY}}(\theta) = -2 \operatorname{trace} C(\theta) + \|V_\alpha(\theta)\|_F^2$$

where $C(\theta)$ is the cross-covariance matrix between the representations of the two views, and $V_\alpha(\theta)$ is a matrix involving the individual covariance matrices of each view.

This objective has two terms: the first term, $-2 \operatorname{trace} C(\theta)$, encourages the representations to be correlated across views, similar to the invariance term in Barlow Twins and VICReg. The second term, $\|V_\alpha(\theta)\|_F^2$, involves the individual covariance matrices, which is analogous to the variance and covariance terms in VICReg. The main difference is that our method is based on canonical correlation principles, which may offer additional benefits in terms of representation quality and interpretability.

In the next section, we will present experiments demonstrating the effectiveness of our method in both the Deep CCA and SSL settings.

subsectionPyTorch Implementation

We provide PyTorch implementations of DCCA-EY and SSL-EY in Listings 1 and 2, respectively.

[1] Given the distributions of both views are identical, there is no reason we would expect assymetric functions to be optimal out of sample

The DCCA-EY implementation defines a class that takes a list of encoders as input and computes the loss function as described in Section 3.2. The forward method computes the representations of the input data using the encoders, while the loss method computes the loss function based on the representations. This implementation can be used to train DCCA-EY models on multi-view data.

The SSL-EY implementation defines a class that takes a single encoder as input, which is used to transform both augmented views of the data. The forward method computes the representations of the input data using the shared encoder, while the loss method computes the loss function based on the representations and the ridge penalty hyperparameters. This implementation can be used to train SSL-EY models on augmented data.

In the next section, we will present experiments demonstrating the effectiveness of our DCCA-EY and SSL-EY methods in their respective settings.

We provide a PyTorch implementation of DCCA-EY in Listing 1. This implementation defines a DCCA-EY class that takes a list of encoders as input and computes the loss function as described in Section 3.2. The forward method computes the representations of the input data using the encoders, while the loss method computes the loss function based on the representations. This implementation can be used to train DCCA-EY models on multi-view data.

4 Experiments and Results

4.1 Deep CCA

In this experiment, we aim to establish the superiority of our DCCA-EY method over existing Deep Canonical Correlation Analysis (DCCA) approaches. We specifically focus on showcasing how DCCA-EY outperforms these methods in terms of correlation capture, convergence speed, and ease of hyperparameter tuning. The experimental setup is aligned with that of W. Wang, Arora, Livescu, and Srebro (2015), providing a direct comparison under identical conditions.

As per W. Wang, Arora, Livescu, and Srebro (2015), our architecture comprises multilayer perceptrons with two hidden layers of size 800 and an output layer of 50 with ReLU activations. We train these networks for 20 epochs. However, our primary goal is to learn $K = 50$ dimensional representations over a range of mini-batch sizes (from 20 to 100) across 50 epochs, demonstrating the robustness and scalability of DCCA-EY even in varying batch conditions.

In this chapter, we employ the Total Correlation Captured (TCC) metric for

```
import torch
import torch.nn as nn

class DCCA_EY(nn.Module):
def init(self, encoders):
super(DCCA_EY, self).init()
self.encoders = nn.ModuleList(encoders)
def forward(self, Xs):
Zs = [encoder(X) for encoder, X in zip(self.encoders, Xs)]
return Zs

def loss(self, Zs):
# Compute total between-view covariance matrix
C = torch.zeros(Zs[0].shape[1], Zs[0].shape[1])
for i in range(len(Zs)):
    for j in range(i + 1, len(Zs)):
        C += torch.matmul(Zs[i].T, Zs[j]) / Zs[i].shape[0]

# Compute total within-view variance matrix
V = torch.zeros(Zs[0].shape[1], Zs[0].shape[1])
for i in range(len(Zs)):
    V += torch.matmul(Zs[i].T, Zs[i]) / Zs[i].shape[0]

# Compute loss
loss = -2 * torch.trace(C) + torch.norm(V, p='fro') ** 2

return loss
```

Listing 1: PyTorch implementation of DCCA-EY.

evaluation. While similar to the PCC metric described in the previous chapter, TCC does not rely on a ground truth for its computation. Instead, it is defined as TCC $= \sum_{k=1}^{K} \rho_k$, where ρ_k are the empirical correlations between the neural network-based representations $Z^{(i)} = f^{(i)}(X^{(i)})$ on a validation set, rather than on the training set as was the case with PCC. This distinction is crucial as TCC evaluates the model's performance in capturing correlations in an unseen dataset, offering a more robust measure of its generalization capability.

4.1.1 Data

The Split MNIST dataset is a modified version of the original MNIST dataset, where each 28x28 pixel grayscale image of handwritten digits (0-9) is divided into left and right halves, creating two distinct views. This split challenges models to learn from

```
    import torch
    import torch.nn as nn
class SSL_EY(nn.Module):
def init(self, encoder):
super(SSL_EY, self).init()
self.encoder = encoder
def forward(self, Xs):
    Zs = [self.encoder(X) for X in Xs]
    return Zs

def loss(self, Zs, alphas=[0, 0]):
    # Compute cross-covariance matrix
    C = torch.matmul(Zs[0].T, Zs[1]) / Zs[0].shape[0]

    # Compute individual covariance matrices
    V1 = torch.matmul(Zs[0].T, Zs[0]) / Zs[0].shape[0]
    V2 = torch.matmul(Zs[1].T, Zs[1]) / Zs[1].shape[0]

    # Compute loss
    V_alpha = alphas[0] * torch.eye(V1.shape[0]) + (1 - alphas[0]) * V1 + \
              alphas[1] * torch.eye(V2.shape[0]) + (1 - alphas[1]) * V2
    loss = -2 * torch.trace(C) + torch.norm(V_alpha, p='fro') ** 2

    return loss
```

Listing 2: PyTorch implementation of SSL-EY.

partial information, as each view contains only half of the digit, either the left or the right side. The dataset comprises 50,000 training and 10,000 test images. The X-Ray Microbeam Speech Production Database (XRMB) is a multi-view dataset used for studying articulatory speech data. It comprises around 40,000 spoken utterances from 47 American English speakers. The dataset provides two views: acoustic features and articulatory measurements. The acoustic features consist of 273-dimensional vectors representing spectral characteristics, while the articulatory measurements include 112-dimensional vectors capturing the position and movement of speech articulators (like the tongue and lips). The XRMB dataset is notable for its complexity and high dimensionality, making it a challenging testbed for multiview learning algorithms.

4.1.2 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
ρ^2	0.6, 0.8, 0.9
epochs	50

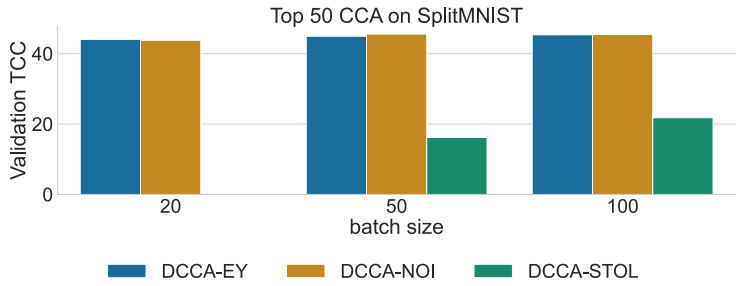


Figure VI.4: Deep CCA on SplitMNIST: Comparison of methods across varying batch sizes.

4.1.3 Observations on SplitMNIST

For the SplitMNIST dataset, Figure VI.4 shows the comparison of methods across different batch sizes. We observe that DCCA-STOL captures significantly less correlation than the other methods and breaks down when the mini-batch size is smaller than the dimension $K = 50$. Figure VI.5 illustrates the learning progress over 50 epochs, where DCCA-NOI, despite performing similarly to DCCA-EY, requires more careful hyperparameter tuning and demonstrates a slower convergence speed.

4.1.4 Observations on XRMB

On the XRMB dataset, as seen in Figure VI.6, similar trends are evident. DCCA-STOL struggles with smaller mini-batch sizes, while DCCA-NOI, though comparable to DCCA-EY in performance, lags in convergence speed, as shown in Figure VI.7.

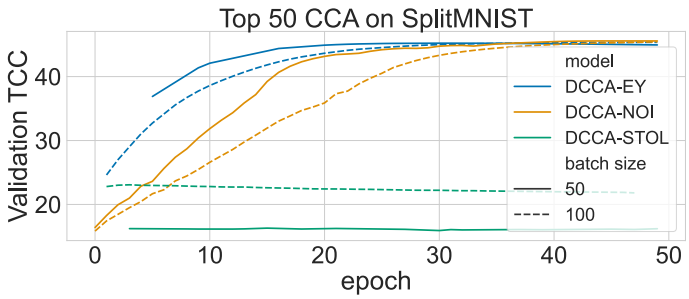


Figure VI.5: Deep CCA on SplitMNIST: Learning progress over 50 epochs.

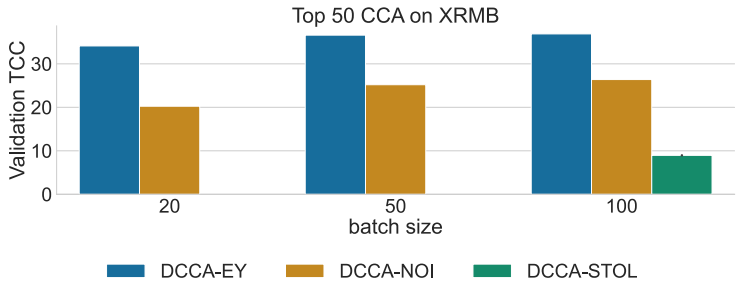


Figure VI.6: Deep CCA on XRMB: Comparison of methods across varying batch sizes.

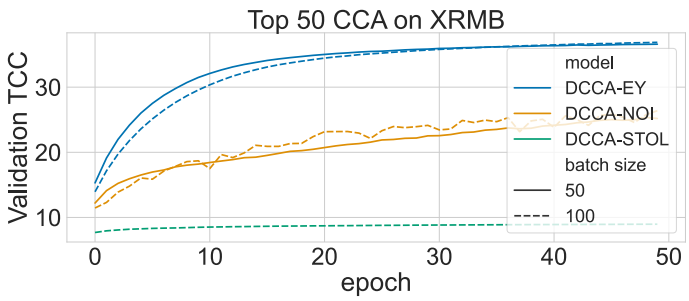


Figure VI.7: Deep CCA on XRMB: Learning progress over 50 epochs.

4.2 Deep Multiview CCA: Robustness Across Different Batch Sizes

In our second experiment, our objective is to showcase the adaptability and effectiveness of the DCCA-EY method in the multiview context, particularly in comparison to existing methods such as DMCCA and DGCCA. We again learn $K = 50$ dimensional representations, but now train for 100 epochs. We employ a multiview extension of the Total Correlation Captured (TCC) metric, termed Total Multiview Correlation Captured (TMCC). TMCC averages the correlation across views and is defined using the consistent notation from Section 2 as:

$$\text{TMCC} = \sum_{k=1}^{K} \frac{1}{I(I-1)} \sum_{\substack{i,j \leq I \\ i \neq j}} \text{corr}(Z_k^{(i)}, Z_k^{(j)}),$$

where $Z_k^{(i)}$ represents the k-th dimension of the i-th view's representation. This metric effectively measures the extent to which our method captures correlations between different views in a multidimensional representation space.

4.2.1 Data

We choose the mfeat dataset for this purpose, which comprises 2,000 handwritten numeral patterns represented through six distinct feature sets, including Fourier coefficients, profile correlations, Karhunen-Love coefficients, pixel averages in 2×3 windows, Zernike moments, and morphological features. These diverse features present an ideal testbed for evaluating the performance of multiview learning methods.

4.2.2 Parameters

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

4.2.3 Observations

Figure VI.8 illustrates the comparison of DCCA-EY with DGCCA and DMCCA across different mini-batch sizes, using the validation TMCC metric. DCCA-EY consistently outperforms both DGCCA and DMCCA, showcasing its superior ability to capture validation TMCC. Notably, DMCCA encounters issues when the batch size is smaller than $K = 50$, likely due to singular empirical covariances. DGCCA, while not

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

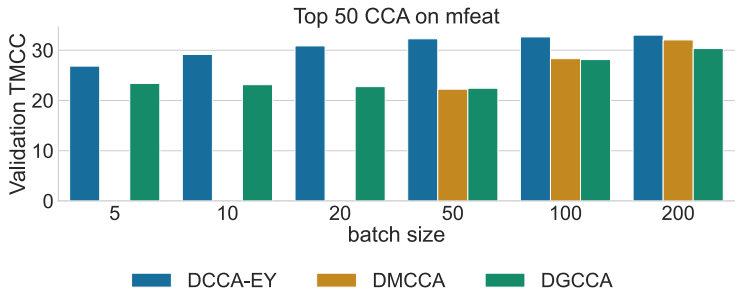


Figure VI.8: Deep Multi-view CCA on mfeat: Comparison across various mini-batch sizes using the Validation TMCC metric.

breaking down, significantly underperforms with smaller batch sizes, highlighting limitations in scalability and efficiency for large-scale data applications.

In Figure VI.9, we observe the learning curves for batch sizes 50 and 100. Both DMCCA and DGCCA demonstrate rapid initial learning of significant correlations but reach a plateau relatively quickly. In contrast, DCCA-EY exhibits a consistent improvement over time and notably outperforms the other methods by the end of the training period. This behavior underscores the enhanced learning capability and efficiency of DCCA-EY, especially in the context of large-scale, high-dimensional data.

4.3 Self-Supervised Learning with SSL-EY

Finally, we benchmark our self-supervised learning algorithm, SSL-EY, with Barlow Twins and VICReg on standard SSL benchmarks. We follow a standard experimental design (Tong et al., 2023). Indeed, we use the sololearn library (Da Costa et al.,

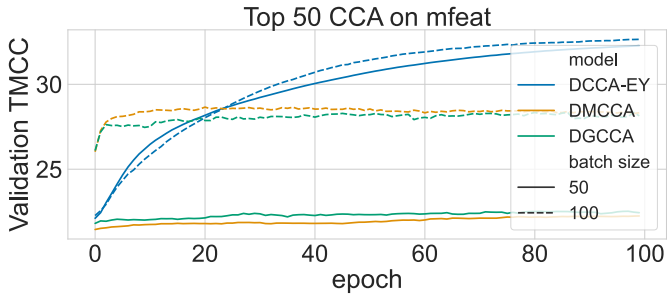


Figure VI.9: Deep Multi-view CCA on mfeat: Learning progress over 100 epochs for batch sizes 50 and 100.

2022), which offers optimized setups particularly tailored for VICReg and Barlow Twins. All methods use 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.

4.3.1 Data

We use the CIFAR-10 and CIFAR-100 datasets, which comprise 60,000 labelled images of size 32x32. CIFAR-10 contains 10 classes, while CIFAR-100 contains 100 classes.

4.3.2 Observations

As Table 4.1 demonstrates, SSL-EY rivals Barlow Twins and VICReg, despite employing general hyperparameters as opposed to the latter's specifically optimized ones.

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: Comparing the performance of SSL methods on CIFAR-10 and CIFAR-100.

4.3.3 Model Convergence

In deep learning, a learning curve usually represents a graph showing the model's learning progress against number of epochs. Figure VI.10 illustrates that the performance variations at 1,000 epochs, shown in Table 4.1, primarily stem from optimization noise, with convergence speeds being comparable among methods.

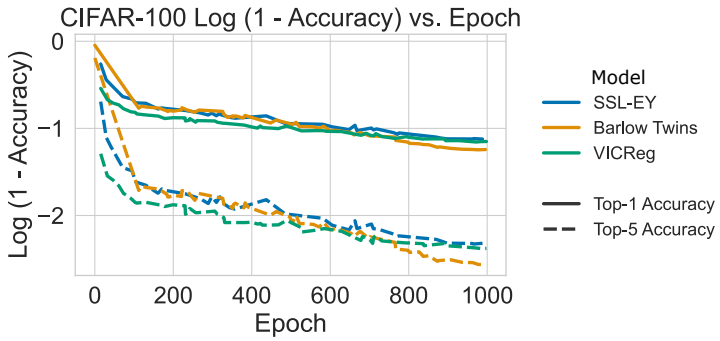
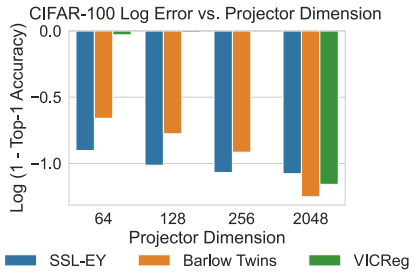


Figure VI.10: Learning curves for SSL-EY, Barlow Twins, and VICReg on CIFAR-100, depicting 1,000-epoch performance.

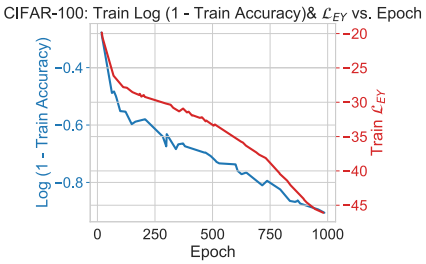
4.3.4 Projector Size Variation

We hypothesized that SSL-EY's robustness to projector size might allow for efficient performance even with smaller projectors or without one. This hypothesis led us to experiment with varying projector output dimensions and completely removing the projector while maintaining the encoder size. Figure VI.11a shows SSL-EY's sustained performance with reduced projector size, indicating more efficient repre-

sentations compared to Barlow Twins and VICReg. Furthermore, as Table 4.1 and Figure VI.11b suggest, SSL-EY performs consistently well even without a projector, underlining its reduced reliance on this architectural component.



(a) Performance of SSL-EY with varying projector dimensions on CIFAR-100.



(b) Correlation of EY loss with classification accuracy in SSL-EY on CIFAR-100.

Figure VI.11: CIFAR 100 Projector Analysis: (a) Examining the impact of projector size on SSL-EY's performance. (b) Investigating the relationship between EY loss and classification accuracy.

4.3.5 $\mathcal{L}_{\text{EY}}$ as an Informative Metric

FigureVI.11b offers two insights. First, it evidences the close relationship between EY loss and classification accuracy, highlighting the potential of maximizing canonical correlation as a pretext task in SSL. Second, it reveals that even a reduced projector dimensionality does not reach full capacity within 1,000 epochs, implying untapped potential in SSL-EY's representation capacity. evolution of the correlation, measured by $\mathcal{L}_{\text{EY}}$, suggests a new avenue for monitoring model training, potentially eliminating the need for a separate validation task.

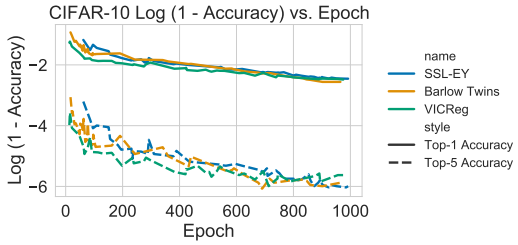
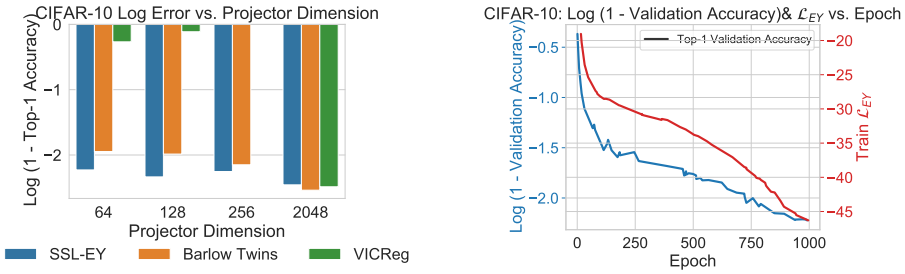


Figure VI.12: Learning curves for SSL-EY, Barlow Twins, and VICReg on CIFAR-10, depicting 1,000-epoch performance.



(a) Performance of SSL-EY with varying projector dimensions on CIFAR-10.

(b) Correlation of EY loss with classification accuracy in SSL-EY on CIFAR-10.

Figure VI.13: CIFAR 10 Projector Analysis: (a) Examining the impact of projector size on SSL-EY's performance. (b) Investigating the relationship between EY loss and classification accuracy.

5 Discussion

5.1 Limitations

Our innovative formulation of DCCA-EY successfully addresses the limitations of traditional DCCA methods, particularly in stochastic settings. The experimental results on Split MNIST and XRMB datasets demonstrate DCCA-EY's superior capability in capturing correlations efficiently across a range of mini-batch sizes, thereby validating its scalability and robustness. Furthermore, our method shows notable improvements in convergence speed and reduced sensitivity to hyperparameter tuning, aspects critically important for practical applications.

In SSL, SSL-EY stands out as a competitive and robust approach. Our experiments on CIFAR-10 and CIFAR-100 highlight SSL-EY's ability to achieve comparable performance with state-of-the-art methods like Barlow Twins and VICReg, even while using general hyperparameters. This is important given the time it takes to run experiments with SSL methods which, as here, can be up to 1,000 epochs. This underscores SSL-EY's adaptability and robustness in different settings. Particularly noteworthy is its performance with reduced or absent projectors, indicating its efficiency and potential for broader applications in SSL. Moreover, the insights gleaned from the correlation of EY loss with classification accuracy in SSL-EY open new avenues for understanding and leveraging canonical correlations in SSL. The observed relationships provide a promising direction for future research in developing more effective and efficient SSL methods.

5.2 Conclusion

Our work bridges significant gaps in the literature and establishes a strong foundation for future explorations in both DCCA and SSL. The proposed methods not only enhance our understanding of these domains but also pave the way for practical applications where scalability, efficiency, and robustness are paramount. In summary, this chapter contributes to the advancement of DCCA and SSL by introducing novel approaches that are not only theoretically sound but also practically viable, offering valuable tools for researchers and practitioners alike in the ever-evolving landscape of machine learning.

Chapter VII

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

Contents

Preface

This work was published in the Journal of Open Source Software (Chapman and H.-T. Wang, 2021). I have been the lead developer of the `CCA-Zoo` package since its inception in 2020. All of the methods we have described in this thesis are implemented in `CCA-Zoo` and are immediately available for use by the research community.

1 Introduction

This chapter presents `CCA-Zoo`, a comprehensive Python library for multiview learning that was developed as a key contribution of this thesis. `CCA-Zoo` brings together a wide range of methods for canonical correlation analysis (CCA), partial least squares (PLS), and related techniques, providing efficient and user-friendly implementations that integrate seamlessly with the Python data science ecosystem.

The development of `CCA-Zoo` was motivated by the recognition that the lack of well-developed and widely available software has been a major barrier to the adoption and advancement of multiview learning methods, particularly in the Python community. While popular libraries like `scikit-learn` (Pedregosa et al., 2011) offer basic implementations of classical techniques like CCA and PLS, they lack support for many of the important extensions and variants that have been proposed in the literature to handle challenges such as high-dimensional data, non-linearity, sparsity, and deep learning.

`CCA-Zoo` aims to fill this gap by providing a unified framework for multiview learning that is both comprehensive and accessible. The library includes implementations of both classical and state-of-the-art methods, ranging from regularized and kernel-based extensions of CCA and PLS to modern deep learning and probabilistic

approaches. These implementations are designed to be efficient, scalable, and easy to use, with a consistent API that follows the conventions of `scikit-learn`.

In addition to its core algorithms, `CCA-Zoo` provides a range of tools and utilities to support the entire multiview learning workflow, from data preprocessing and feature selection to model evaluation and visualization. The library also includes a collection of example datasets and pre-trained models, making it easy for users to get started and explore different techniques on real-world problems.

Throughout the development of this thesis, `CCA-Zoo` has played a central role as both a research tool and a means of disseminating our methodological contributions to the wider community. The experiments and case studies presented in the previous chapters have all relied on `CCA-Zoo` implementations, ensuring reproducibility and comparability of our results. At the same time, by releasing `CCA-Zoo` as an open-source library on GitHub and PyPi, we have enabled other researchers and practitioners to easily build upon and extend our work.

We also discuss the impact that `CCA-Zoo` has had so far, both within the context of this thesis and in the broader research community, and outline directions for future development and improvement. Our hope is that `CCA-Zoo` will serve as a valuable resource and catalyst for advancing the state-of-the-art in multiview learning, and for bridging the gap between methodological research and practical application.

2 Background

The field of multiview learning has recently witnessed a surge of interest from the research community. This growth can be attributed to the increasing availability of multi-modal data across various domains, from bioinformatics to social media analysis, and the recognition that integrating multiple views can often lead to better insights and predictions than relying on a single perspective.

Traditionally, the development of multiview learning methods has been dominated by researchers in the statistical learning community, who have primarily relied on programming languages like R and MATLAB. These platforms have served as fertile ground for the creation and dissemination of many state-of-the-art algorithms.

However, this has created a challenge for researchers and practitioners who prefer to work in the Python programming language, which has become increasingly popular for machine learning tasks due to its simplicity, flexibility, and rich ecosystem of libraries. Python users have been faced with two suboptimal options: either port existing R or MATLAB implementations into Python, which can be a time-consuming

and error-prone process requiring significant domain expertise, or make do with the limited set of multiview methods available in general-purpose Python libraries like `scikit-learn` (Pedregosa et al., 2011).

This fragmentation of the multiview learning landscape across different programming languages has created significant barriers to entry for Python users, potentially hindering the widespread adoption and application of these powerful techniques. Moreover, it has made it difficult for researchers to compare and benchmark different methods on a level playing field, as implementations may vary widely in terms of performance, scalability, and ease of use.

The `CCA-Zoo` package aims to address these challenges by providing a comprehensive and unified platform for multiview learning in Python. By offering a wide range of algorithms spanning both classical and state-of-the-art approaches, `CCA-Zoo` enables researchers and practitioners to easily explore and apply these techniques to their own data and problems, without the need for extensive domain expertise or cumbersome porting of code.

Through its scikit-learn compatible API, modular design, and efficient implementations, `CCA-Zoo` seamlessly integrates with the existing Python machine learning ecosystem, lowering the barriers to entry and accelerating the pace of research and application in multiview learning. By bringing together methods from different research communities and programming languages under a common framework, `CCA-Zoo` also facilitates fair and reproducible comparisons of different approaches, helping to advance our understanding of their strengths and limitations.

In the following sections, we will consider the design principles, key features, and implementation details of `CCA-Zoo`, showcasing how it can be used to streamline and enhance multiview learning workflows in Python.

3 Methods

`CCA-Zoo` is designed to be a comprehensive and user-friendly library for multiview learning in Python. In this section, we describe the key design decisions and implementation details that underpin its functionality, flexibility, and performance. Figure VII.1 provides an overview of the library's structure and its integration with the wider Python machine learning ecosystem.

3.1 API Design and Scikit-Learn Compatibility

A central goal in the development of `CCA-Zoo` was to ensure maximum compatibility and interoperability with the existing Python machine learning ecosystem. To this end, we adopted the API design principles and conventions of the widely-used `scikit-learn` library (Pedregosa et al., 2011). `Scikit-learn` has become the de facto standard for machine learning in Python, thanks to its consistent, user-friendly API and its extensive collection of tools for data preprocessing, model selection, evaluation, and visualization.

By adhering to the `scikit-learn` API, `CCA-Zoo` inherits these benefits and ensures that users can seamlessly integrate multiview learning methods into their existing workflows. All estimators in `CCA-Zoo` follow the fit-transform pattern, where the `fit()` method learns model parameters from training data, and the `transform()` method applies the learned transformation to new data. Hyperparameters are specified as constructor arguments, allowing easy model creation and configuration.

This design choice not only makes `CCA-Zoo` intuitive and easy to use for anyone familiar with `scikit-learn`, but also enables the use of `scikit-learn`'s powerful model selection and evaluation tools, such as cross-validation and grid search, directly with `CCA-Zoo` estimators. Users can construct complex pipelines that include data preprocessing, feature selection, and multiview learning steps, all with a consistent, declarative syntax.

Figure VII.1 illustrates this integration, highlighting `CCA-Zoo`'s compatibility with key components of the Python machine learning stack.

3.2 Modular Architecture and Extensibility

Another key design principle of `CCA-Zoo` is modularity. The library is organized into distinct submodules, each focusing on a specific aspect of the multiview learning workflow:

- `datasets`: Classes for generating synthetic data and loading real-world datasets.
- `preprocessing`: Tools for data normalization, scaling, and dimensionality reduction.
- `model_selection`: Wrappers for `scikit-learn`'s cross-validation and hyperparameter tuning utilities, adapted for multiview settings.
- `linear`: Estimators for linear CCA, PLS, and their variants.

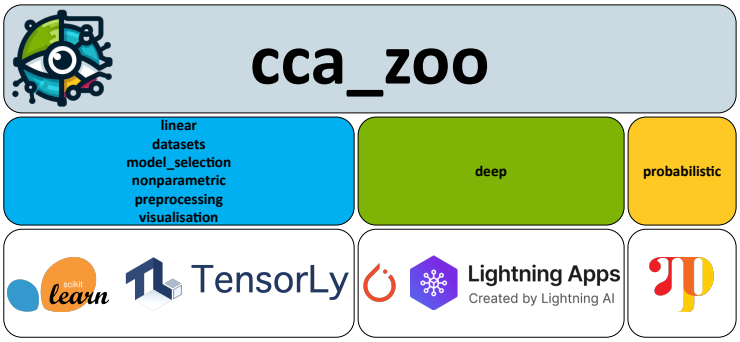


Figure VII.1: The `CCA-Zoo` compatibility map showcases integration with various machine learning packages. The deep learning module is built upon `PyTorch` and `Lightning`, reflecting their status as industry standards for neural network implementations. The probabilistic module employs `NumPyro` for its Bayesian inference capabilities, enhancing the application of probabilistic approaches in CCA.

- `deep`: Deep learning-based approaches, built on top of `PyTorch` (Paszke et al., 2019) and `PyTorch Lightning` (Falcon, 2019).
- `probabilistic`: Bayesian multiview learning methods, implemented with `NumPyro` (Phan, Pradhan, and Jankowiak, 2019), a probabilistic programming library built on top of `JAX` (Babuschkin et al., 2020).
- `visualization`: Functions for visualizing model parameters, latent spaces, and performance metrics.

This modular structure makes the codebase more maintainable and easier to navigate. It also facilitates extensibility: new methods and features can be added to each submodule without affecting the rest of the library, as long as they adhere to the common API conventions.

Furthermore, the use of well-established libraries like `PyTorch`, `PyTorch Lightning`, and `NumPyro` for the deep learning and probabilistic modules ensures that `CCA-Zoo` can benefit from the latest advancements in these rapidly evolving fields. Developers can easily experiment with new architectures, loss functions, and inference techniques, while still leveraging the data handling and model evaluation capabilities of the core `CCA-Zoo` framework.

3.3 Flexibility and Ease of Use

`CCA-Zoo` is designed to be flexible and easy to use for a wide range of multiview learning tasks. The library provides a unified interface for working with both linear and nonlinear methods, unsupervised and semi-supervised settings, and two-view and multi-view scenarios.

The choice of default hyperparameters and architectural choices for deep learning models has been carefully considered to ensure good performance on a variety of datasets without the need for extensive tuning. At the same time, users have full control over these settings and can easily customize them for specific tasks.

`CCA-Zoo` also includes a range of utility functions and classes that simplify common tasks and help users avoid boilerplate code. For example, the `datasets` module provides a consistent interface for accessing and sampling from both synthetic and real-world datasets, handling data loading, splitting, and formatting behind the scenes.

Similarly, the `model_selection` module extends `scikit-learn`'s cross-validation and grid search tools to handle the multi-view setting seamlessly. Users can perform model selection and hyperparameter tuning with just a few lines of code, without having to worry about the intricacies of indexing and reshaping views.

Listing 3 demonstrates this simplicity and flexibility, showing a complete workflow for training and evaluating a regularized CCA model with cross-validated hyperparameter selection:

This example showcases several key features of `CCA-Zoo`:

- The `LatentVariableData` class allows easy generation of synthetic multiview data with a specified number of features and latent dimensions.
- The `rCCA` class provides a regularized CCA estimator with a `scikit-learn`-compatible API, supporting both fit-transform and inverse transform operations.
- The `GridSearchCV` class wraps `scikit-learn`'s grid search functionality, automatically handling the multi-view parameter grid and cross-validation splitting.
- The `SeparateRepresentationScatterDisplay` class provides a high-level interface for visualizing the learned latent space, with separate plots for each view.

```
from cca_zoo.datasets import LatentVariableData
from cca_zoo.linear import rCCA
from cca_zoo.model_selection import GridSearchCV
from cca_zoo.visualisation import SeparateRepresentationScatterDisplay

#Generate synthetic multi-view data
data = LatentVariableData(view_features=[10, 10], latent_dims=2)
X, Y = data.sample(n_samples=200, seed=42)

#Define hyperparameter grid
param_grid = {
'c': ([0.1, 0.3, 0.7, 0.9], [0.1, 0.3, 0.7, 0.9]),
}

#Perform cross-validated grid search
model = GridSearchCV(rCCA(latent_dimensions=2),
param_grid=param_grid,
cv=5).fit(X, Y)

#Visualize latent space
SeparateRepresentationScatterDisplay.from_estimator(model.best_estimator_)
```

Listing 3: A complete example of training and evaluating a regularized CCA model with `CCA-Zoo`.

By providing such high-level abstractions and adhering to familiar API conventions, `CCA-Zoo` aims to make multiview learning methods accessible to a wide audience, from seasoned machine learning practitioners to domain experts in fields like bioinformatics, computer vision, and natural language processing.

3.4 Performance and Scalability

In addition to ease of use and flexibility, `CCA-Zoo` is designed with performance and scalability in mind. The library is implemented in pure Python, with computationally intensive operations delegated to optimized libraries like `NumPy` (Harris et al., 2020), `SciPy` (Virtanen et al., 2020), and `PyTorch`.

For linear methods like CCA and PLS, `CCA-Zoo` leverages the randomized SVD and other matrix approximation techniques to efficiently handle high-dimensional data. These techniques allow the library to scale to datasets with millions of features and samples, without sacrificing accuracy or numerical stability.

In the deep learning module, `CCA-Zoo` takes advantage of PyTorch's GPU accel-

eration and automatic differentiation capabilities to enable fast training of complex models on large-scale datasets. The use of `PyTorch Lightning` further streamlines the training process, providing a high-level interface for distributed training, checkpointing, and logging.

For probabilistic methods, `CCA-Zoo` leverages the power of `NumPyro` and `JAX` to perform efficient variational inference and MCMC sampling on both CPUs and GPUs. The use of modern probabilistic programming techniques allows users to easily specify and train complex Bayesian models, while still benefiting from the performance and scalability of the underlying libraries.

`CCA-Zoo`'s performance and scalability claims are backed by extensive benchmarking and testing on a variety of synthetic and real-world datasets. In Section 4, we present a selection of these experiments, comparing `CCA-Zoo`'s performance to other popular multiview learning libraries and demonstrating its ability to handle large-scale, high-dimensional data.

3.5 Development and Maintenance

`CCA-Zoo` is developed as an open-source project, with its source code and documentation hosted on GitHub at `https://github.com/jameschapman19/cca_zoo`. The library follows modern software development best practices, including version control, continuous integration, and automated testing.

The development team is committed to maintaining and improving `CCA-Zoo` over the long term. This includes fixing bugs, adding new features and methods, and keeping dependencies up to date. The team also welcomes contributions from the community in the form of bug reports, feature requests, and pull requests.

To ensure the library's quality and reliability, `CCA-Zoo` includes a comprehensive test suite that covers all major functionality. These tests are automatically run on each commit and pull request, using continuous integration services like Travis CI and GitHub Actions. This helps catch regressions and ensures that new features are properly integrated and documented.

`CCA-Zoo`'s documentation is another key aspect of its maintenance and development. The library includes extensive API documentation, generated automatically from docstrings using tools like Sphinx and Read the Docs. The documentation also includes user guides, tutorials, and examples to help users get started and make the most of the library's features.

In addition to the API documentation, `CCA-Zoo`'s GitHub repository includes a wiki and issue tracker where users can find additional information, ask questions,

and report bugs. The development team is responsive to user feedback and strives to address issues in a timely manner.

By adhering to these development and maintenance practices, `CCA-Zoo` aims to provide a stable, reliable, and well-documented library that can serve as a foundation for multiview learning research and applications for years to come.

In summary, the key design decisions and implementation details of `CCA-Zoo` are:

- Adherence to the `scikit-learn` API for maximum compatibility and ease of use.
- Modular architecture for maintainability and extensibility.
- Flexibility and unified interface for both linear and deep learning methods.
- Use of optimized libraries and techniques for performance and scalability.
- Open-source development with modern software engineering practices.
- Comprehensive documentation and user support.

These choices reflect `CCA-Zoo`'s goal of providing a powerful yet accessible toolkit for multiview learning in Python, suitable for both research and practical applications.

Here's an improved version of the Experiments section that highlights the use of CCA-Zoo throughout the thesis and includes additional benchmarking details:

4 Experiments

Throughout this thesis, the `CCA-Zoo` package has been used extensively for conducting experiments and evaluating the proposed multiview learning methods. The library's comprehensive set of tools and its seamless integration with the Python data science ecosystem have greatly facilitated the implementation and assessment of these methods on a wide range of datasets and tasks.

In this section, we showcase the performance and versatility of `CCA-Zoo` through a series of benchmarking experiments. These experiments not only demonstrate the efficiency of the library's implementations but also highlight its ability to handle high-dimensional data, a crucial requirement in many real-world applications such as bioinformatics and natural language processing.

4.1 Benchmarking Setup

To assess the computational efficiency of `CCA-Zoo`, we compared its performance against the widely-used `scikit-learn` library. We focused on two fundamental multiview learning methods: Canonical Correlation Analysis (CCA) and Partial Least Squares (PLS). The experiments were conducted on synthetic datasets of varying dimensionality to evaluate the scalability of the implementations.

The datasets were generated as random matrices with a fixed number of samples (100) and a varying number of features (50, 100, 200, 400, and 800) for each view. The number of latent dimensions was set to 10 for both CCA and PLS. To obtain reliable performance metrics, each experiment was repeated 10 times, and the average execution time was reported.

The benchmarking experiments were performed using the following software versions:

- `CCA-Zoo` (version: 2.4.0)
- `Scikit-learn` (version: 1.3.0)

All experiments were run on a machine with an Intel Core i7-9700K CPU (3.60GHz) and 32GB of RAM, running Ubuntu 20.04.

4.2 Canonical Correlation Analysis

Figure VII.2 presents the comparison of execution times between `CCA-Zoo` and `scikit-learn` for CCA. Across all tested dimensionalities, `CCA-Zoo` demonstrates competitive performance, with execution times comparable to or better than those of `scikit-learn`.

The efficiency of `CCA-Zoo`'s CCA implementation can be attributed to its use of the principal component space for computing the canonical correlations. By first projecting the data onto a lower-dimensional space defined by the leading principal components, `CCA-Zoo` reduces the computational burden associated with high-dimensional covariance matrices, resulting in faster execution times without sacrificing accuracy.

This performance advantage is particularly relevant in real-world applications, where the number of features often greatly exceeds the number of samples. In such scenarios, `CCA-Zoo`'s ability to efficiently handle high-dimensional data can lead to significant time savings and enable the analysis of larger datasets.

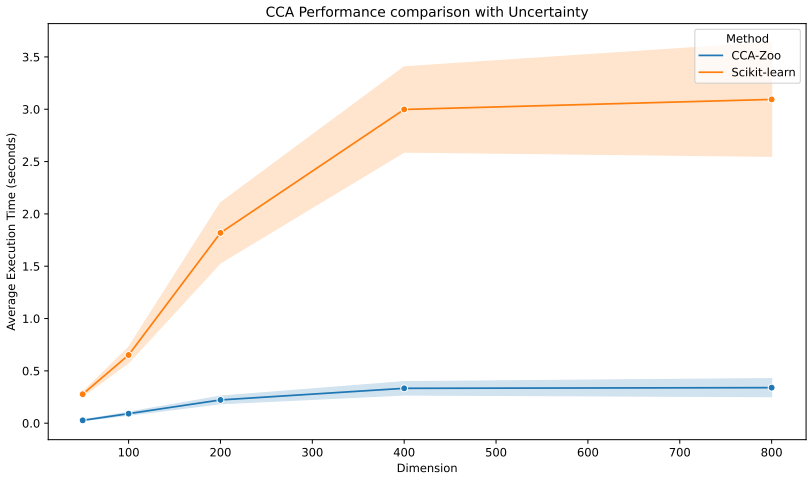


Figure VII.2: Performance comparison for CCA methods

4.3 Partial Least Squares

Figure VII.3 shows the execution time comparison for PLS. Similar to the CCA results, `CCA-Zoo` exhibits a robust performance profile, with execution times that are competitive with those of `scikit-learn` across all tested dimensionalities.

The competitive performance of `CCA-Zoo`'s PLS implementation can be attributed to its use of efficient algorithms and data structures, such as the NIPALS algorithm and the deflation scheme for computing the latent components. These optimizations allow `CCA-Zoo` to scale well with increasing data dimensionality, making it a suitable choice for a wide range of applications.

4.4 Real-World Applications

In addition to the synthetic benchmarking experiments, `CCA-Zoo` has been used throughout this thesis to evaluate the proposed multiview learning methods on various real-world datasets. These datasets span multiple domains, including bioinformatics and computer vision and pose diverse challenges in terms of dimensionality, sparsity, and noise.

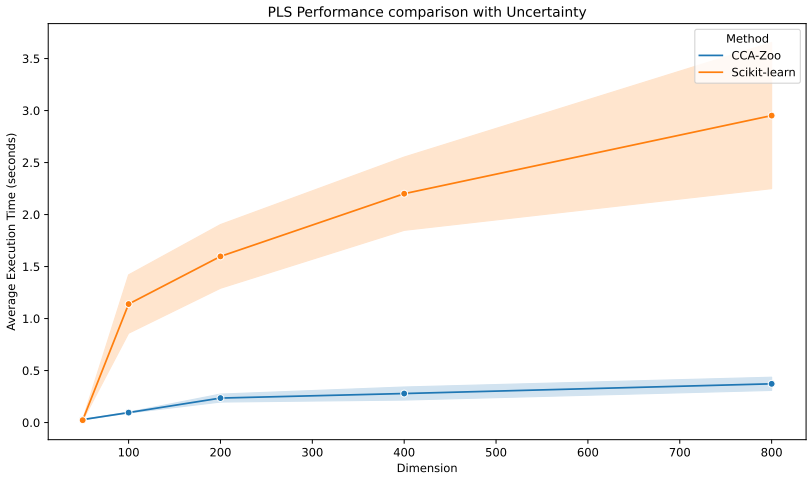


Figure VII.3: Performance comparison for PLS methods

5 Discussion

5.1 Contributions and Impact

The development of CCA-Zoo represents a significant contribution to the field of multiview learning, particularly within the Python ecosystem. By providing a comprehensive and user-friendly library that integrates seamlessly with existing tools like scikit-learn and PyTorch, CCA-Zoo has the potential to greatly accelerate research and application of multiview methods.

Throughout this thesis, CCA-Zoo has played a central role in enabling the empirical studies and method development presented in the previous chapters. The library's efficient implementations of both classical and state-of-the-art multiview algorithms allowed us to conduct extensive experiments on real-world datasets, comparing the performance of different methods and gaining new insights into their behavior. Moreover, the modular design of CCA-Zoo facilitated rapid prototyping and testing of novel extensions and refinements to existing techniques.

Beyond its use in this thesis, CCA-Zoo has already begun to have an impact in the wider research community. The library has been well-received on GitHub, with over 150 stars and 30 forks to date, and has been downloaded nearly 500 times per month from the Python Package Index. Several published papers and ongoing projects in fields ranging from genomics to neuroscience have utilized CCA-Zoo,

demonstrating its potential to enable new discoveries and applications.

5.2 Limitations and Future Work

While CCA-Zoo provides a solid foundation for multiview learning in Python, there are certainly areas where it could be improved and extended. One current limitation is the lack of GPU acceleration for some of the more computationally intensive methods, which could hamper their scalability to massive datasets. In future versions, we plan to leverage libraries like cupy to enable seamless GPU support.

Another direction for future work is to expand the library's functionality to encompass an even wider range of multiview learning paradigms, such as multi-modal deep learning, multi-view clustering, and multi-view matrix factorization. By providing a unified interface to these diverse approaches, CCA-Zoo could serve as a powerful toolkit for exploring and combining different perspectives on data.

Finally, we are committed to the ongoing maintenance and development of CCA-Zoo as an open-source project. We welcome contributions from the community in the form of bug reports, feature requests, documentation improvements, and code contributions. By engaging with users and incorporating their feedback, we hope to continuously refine and enhance the library to better serve the needs of multiview learning researchers and practitioners.

6 Conclusion

In conclusion, CCA-Zoo fills an important gap in the Python ecosystem by providing a comprehensive, efficient, and user-friendly library for multiview learning. Through its extensive catalog of classical and modern multiview methods, seamless integration with popular machine learning tools, and flexible API, CCA-Zoo enables researchers and practitioners to easily explore and apply these powerful techniques to their own data and problems.

The development of CCA-Zoo has been a central contribution of this thesis, underpinning many of the empirical studies and methodological advances presented in earlier chapters. By making the library open-source and freely available to the community, we hope to accelerate progress in multiview learning and promote reproducible, extensible research.

Looking ahead, we see ample opportunities to expand and refine CCA-Zoo, in collaboration with its growing base of users and contributors. Through sustained development and community engagement, we believe CCA-Zoo has the potential

to become an indispensable tool in the multiview learning toolkit, enabling new discoveries and applications across a wide range of domains.

Chapter VIII

Conclusion

This chapter provides a summary of the findings of this thesis, discusses their implications, and outlines potential directions for future work.

1 Summary of Contributions

1.1 Regularisation of CCA Models: A Flexible Framework based on Alternating Least Squares

This chapter presented the FRALS framework for CCA, addressing challenges in analyzing large-scale neuroimaging datasets from projects such as the Human Connectome Project and the Alzheimer's Disease Neuroimaging Initiative. Incorporating structured priors through regularization, particularly the elastic net penalty, FRALS enhanced the interpretability and generalizability of CCA models. This method, documented in the work presented at the OHBM (James Chapman, 2023), has been effective in uncovering significant brain-behavior associations, showing superior out-of-sample performance compared to traditional methods.

1.2 Insights From Generating Simulated Data for CCA

This chapter contributed to the debate on the interpretation of model weights versus loadings in CCA. By generating high-dimensional simulated data and categorizing methods into explicit and implicit latent variable models, the chapter highlights the robustness of loadings to columnwise transformations in data matrices, a feature not shared with weights. The simulated data strategies formed part of the analysis

in Mihalik, Chapman, Rick A Adams, et al. (2022a) and influenced the analysis in Rick A. Adams et al. (2024).

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

Focusing on scaling challenges for CCA and PLS in the context of large-scale biomedical datasets like the UK Biobank, this chapter introduces a new gradient descent algorithm tailored for generalized eigenvalue problems. The methods developed, informed by publications (Chapman, Aguila, and Wells, 2022; Chapman, Wells, and Aguila, 2024; Chapman and Wells, 2023), enable the application of multiview CCA and PLS to datasets with extensive dimensions and complex structures.

1.4 Deep CCA and Self-Supervised Learning

This chapter introduces a novel formulation of Deep CCA optimized for the stochastic minibatch setting and proposes SSL-EY, a new competitive SSL method. Grounded in findings from (Chapman and Wells, 2023) and (Chapman, Wells, and Aguila, 2024), the chapter demonstrates the robustness of these methods against hyperparameter sensitivity and elucidates connections between CCA-based SSL methods and other contemporary SSL approaches.

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

Presenting `CCA-Zoo`, a Python library that consolidates and enhances the accessibility of multiview learning methods, this chapter details the development and capabilities of the library, which implements a variety of CCA, PLS, and related techniques. As detailed in (Chapman and H.-T. Wang, 2021), `CCA-Zoo` addresses gaps in existing software offerings and facilitates broader adoption and innovation within the research community.

In summary, we have demonstrated novel ways to introduce structured priors into CCA models, developed efficient algorithms for large-scale CCA, extended CCA to deep learning, and provide a unified interface for various CCA methods. Finally, we have made software implementations of these methods available to the research community through the `CCA-Zoo` package which have already been well-received by the community.

2 Future Work

2.1 Applications

2.1.1 Large-Scale Neuroimaging Datasets

While the applications presented in this thesis, particularly the UK Biobank analysis in Chapter VI, have demonstrated the potential of our methods, there is still vast untapped potential in applying these techniques to even larger and more diverse datasets. The ABCD dataset, for instance, offers a rich source of multimodal data that could benefit from the regularized and scalable CCA methods developed in this thesis. Preliminary results on this dataset have shown promise, and we believe that further exploration will yield valuable insights into brain development and its associated factors.

2.1.2 Wearable Devices

The rise of wearable devices and the proliferation of biometric data present new opportunities for applying multiview learning techniques to personal health monitoring. By integrating data streams from devices such as smartwatches, continuous glucose monitors, and sleep trackers, we can gain insights into an individual's physical and mental well-being that were previously inaccessible. I strongly believe that the development of interpretable and scalable methods for analyzing these diverse data sources will be crucial for unlocking the full potential of wearable technology in personalized healthcare.

2.2 Methods

2.2.1 Proximal Gradient Descent for Regularized CCA

Preliminary experiments suggest that the proximal gradient descent approach is much faster than existing methods, making it a promising direction for future research. We anticipate that this methodology will significantly enhance the scalability and applicability of CCA, PCA, and PLS in the era of big data and complex regularization schemes.

3 Closing Remarks

My PhD journey has been a fascinating exploration of the world of multiview learning, with Canonical Correlation Analysis (CCA) at its core. What began as a quest to apply deep learning to uncover brain-behavior associations quickly evolved into a multifaceted endeavor that led me to develop scalable algorithms for linear CCA and investigate the connections between CCA and self-supervised learning.

When I first embarked on this journey, I was eager to apply Deep CCA to high-dimensional neuroimaging data to gain insights into the complex relationship between the brain and mental health. However, I soon realized that existing Deep CCA methods were computationally infeasible for such datasets, and even moderately sized datasets could not fully capture the intricacies of this relationship.

Undeterred, I shifted my focus to developing efficient ways to regularize CCA models. This led me to confront the scalability bottleneck of CCA head-on. By developing efficient algorithms for CCA and Partial Least Squares (PLS), I was able to push the boundaries of what was possible with these methods. This exploration also led me to investigate the connections between CCA and self-supervised learning, bringing my journey full circle back to the realm of deep learning.

One of the highlights of my journey has been witnessing the rapid growth and evolution of multiview learning during my PhD. From the emergence of powerful multimodal language models to the increasing adoption of self-supervised learning techniques, it has been thrilling to be a part of this dynamic and fast-paced field.

Another highlight has been the impact of the software we developed, such as the CCA-Zoo package. Seeing researchers across various fields utilize our tools to tackle a wide range of problems has been immensely gratifying. It is a testament to the importance of developing accessible and efficient implementations of these methods.

As I reflect on this journey, I am filled with excitement for the future of multiview learning. The integration of deep learning with CCA and the application of these methods to ever-larger and more diverse datasets hold immense promise. I believe that the work presented in this thesis has laid a foundation for further advancements in this field, and I am eager to see how others will build upon it.

Thank you for reading.

Appendices

Appendix A

HCP and ADNI Loadings

This appendix builds upon the results presented in Chapter III, where we introduced a method to regularize CCA using structured priors on model weights, demonstrated with Human Connectome Project (HCP) and Alzheimer's Disease Neuroimaging Initiative (ADNI) data. In light of the insights gained from Chapter IV, which examined the relationship between loadings and weights in CCA using simulated data, we revisit the HCP and ADNI results to further explore the interpretability of the models.

1 Human Connectome Project (HCP) Data

1.1 Brain Connectivity Weights and Loadings

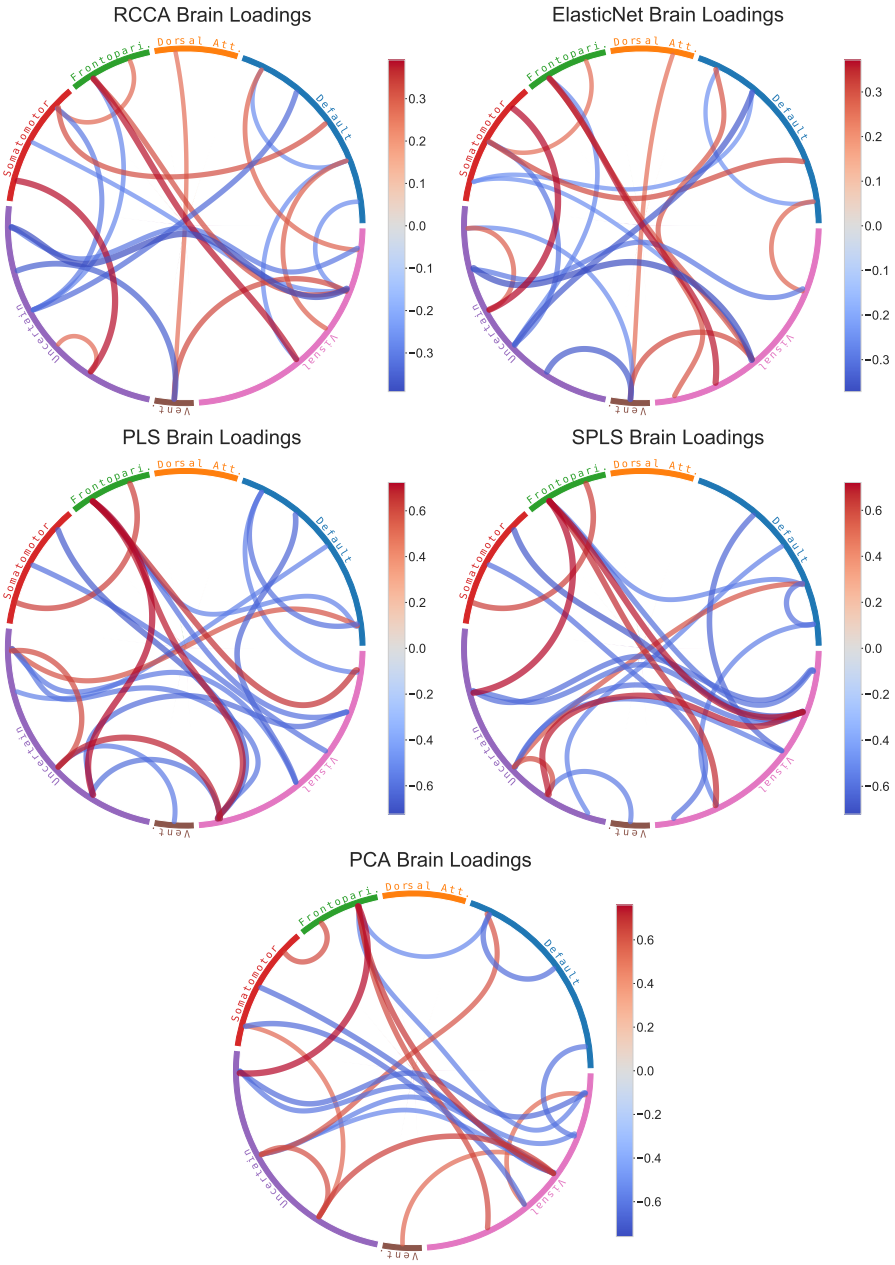


Figure A.1: Chord diagrams of the top 8 positive and negative brain loadings for each model.

2 Alzheimer's Disease Neuroimaging Initiative (ADNI) Data

2.1 Brain Structure Weights and Loadings

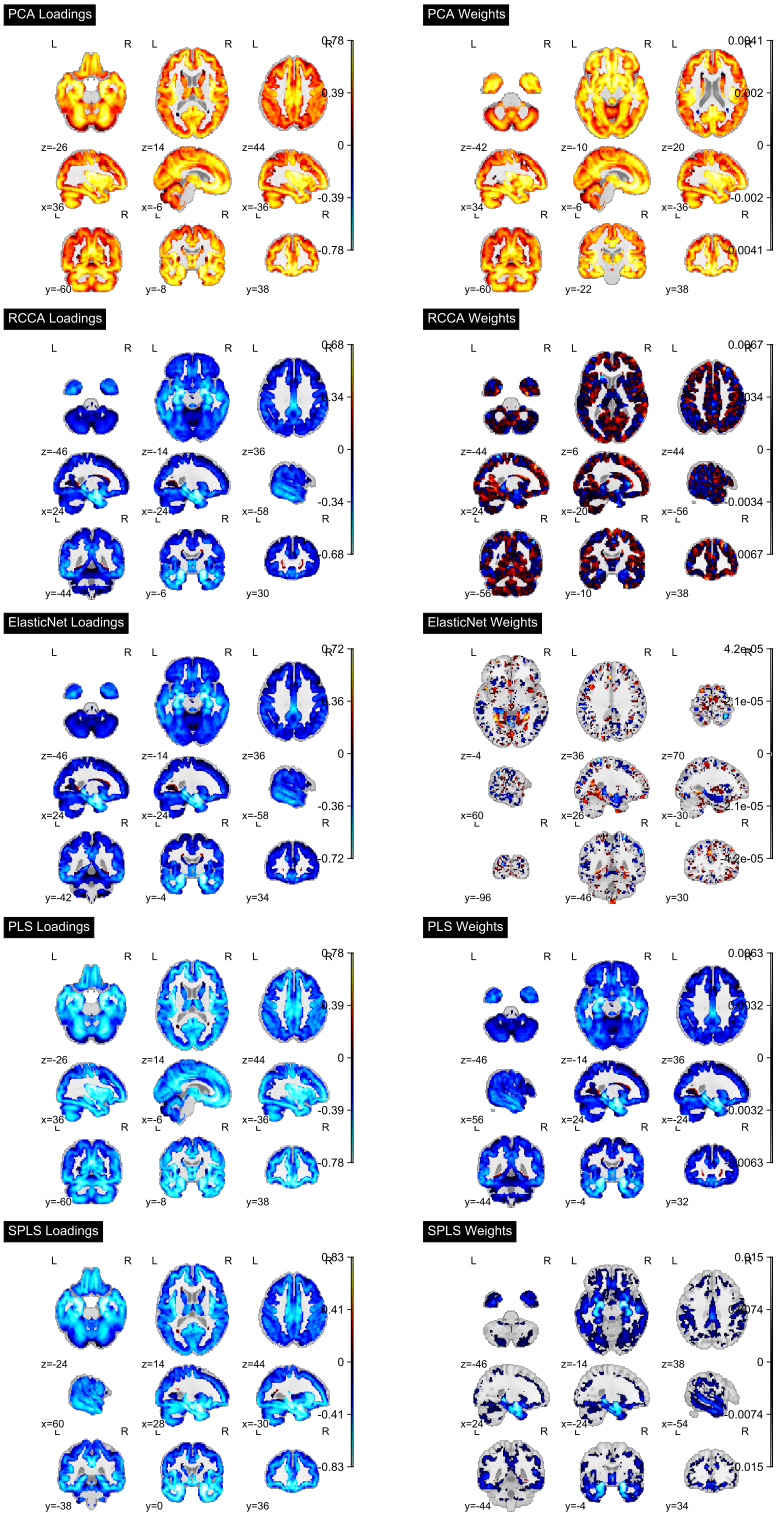


Figure A.2: Statistical maps of brain structure loadings and weights for each model.

Appendix B

Proofs and Additional Results for Chapter V

1 Eckhart-Young characterization of GEP subspace

1.1 Formal definitions

There are various different notations and conventions for GEPs and SVDs. We largely follow the standard texts on Matrix Analysis (Stewart and J.-G. Sun, 1990; R. Bhatia, 1997) but seek a more careful handling of the equality cases of certain results. To help, we use the following non-standard definitions, largely inspired by Carlsson (2021).

Definition 1.1 (Top-K subspace)**.** *Let the GEP (A, B) on $\mathbb{R}^d$ have eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d$. Then **a** top-$K$ subspace is that spanned by some $w_1, \ldots, w_K$, where w_k is a λ_i-eigenvector of (A, B) for $k = 1, \ldots, K$.*

Definition 1.2 (B-orthonormality)**.** *Let $B \in \mathbb{R}^{d \times d}$ be strictly positive definite. Then we say a collection $w_1, \ldots, w_K \in \mathbb{R}^d$ of vectors is B-orthonormal if $w_k^T B w_l = \delta_{kl}$ for each $k, l \in \{1, \ldots, K\}$.*

Definition 1.3 (Top-K matrix)**.** *We say $W \in \mathbb{R}^{d \times K}$ is a top-K matrix for a GEP (A, B) if the k^{th} column w_i of W is a λ_k-eigenvector for each k and the columns are B-orthonormal.*

1.2 Standard Eckhart–Young inequality

Theorem 1.1 (Eckhart–Young)**.** *Let $M \in \mathbb{R}^{p\times q}$. Then $\hat{M}$ minimises $\|M-\tilde{M}\|_F$ over matrices $\tilde{M}$ of rank at most K if and only if $\hat{M} = A_K R_K B_K^\top$ where (A_K, R_K, B_K) is some top-K SVD of the target M.*

Proof. Let $M, \tilde{M}$ have singular values $\sigma_k, \tilde{\sigma}_k$ respectively. Since $\tilde{M}$ has rank at most K we must have $\tilde{\sigma}_k = 0$ for $k > K$.

Then by von Neumann's trace inequality (Carlsson, 2021),

$$\langle M, \tilde{M}\rangle_F \leq \sum_{k=1}^{K} \sigma_k \tilde{\sigma}_k$$

with equality if and only if $M, \tilde{M}$ 'share singular vectors'; the notion of sharing singular vectors is defined as in Carlsson (2021) and in this case means that $\tilde{M} = A_K \tilde{R}_K B_K$ where (A_K, R_K, B_K) is some top-K SVD of M and $\tilde{R}_K$ is a diagonal matrix with decreasing diagonal elements $\tilde{\sigma}_1 \geq \cdots \geq \tilde{\sigma}_K$.

Expanding out the objective and applying this inequality gives

$$\begin{aligned}
\|\tilde{M} - M\|_F^2 &\geq \sum_{k=1}^{d} \sigma_k^2 - 2\sum_{k=1}^{K} \sigma_k \tilde{\sigma}_k + \sum_{k=1}^{K} \tilde{\sigma}_k^2 \\
&= \sum_{k=K+1}^{d} \sigma_k^2 + \sum_{k=1}^{K} (\sigma_k - \tilde{\sigma}_k)^2 \\
&\geq \sum_{k=K+1}^{d} \sigma_k^2
\end{aligned}$$

so indeed to have equality in both cases requires $\sigma_k = \tilde{\sigma}_k$ for each $k \leq K$ so indeed $\tilde{R}_K = R_K$ and so $\hat{M}$, as defined in the statement of the theorem, minimises $\|M - \tilde{M}\|_F$ over matrices $\tilde{M}$ of rank at most K. □

1.3 Supporting Results

Lemma 1.1 (Matrix square root lemma)**.** *Suppose we have two full rank matrices $E, F \in \mathbb{R}^{d\times K}$ where $K \leq d$ and such that $EE^T = FF^T$; then there exists an orthogonal matrix $O \in \mathbb{R}^{K\times K}$ with $E = FO$.*

Proof. Post multiplying the defining condition gives $EE^TE = FF^TE$. Then right

multiplying by $(E^TE)^{-1}$ gives

$$E = FF^TE(E^TE)^{-1} =: FO$$

to check that O as defined above is orthogonal we again use the defining condition to compute

$$O^TO = (E^TE)^{-1}E^TFF^TE(E^TE)^{-1} = (E^TE)^{-1}E^TEE^TE(E^TE)^{-1} = I_K$$

□

Corollary 1.1 (PSD Eckhart–Young for square root matrix)**.** *Let $M \in \mathbb{R}^{d\times d}$ be symmetric positive semidefinite. Then*

$$\underset{\tilde{Z}\in\mathbb{R}^{d\times K}}{\arg\min} \|M - \tilde{Z}\tilde{Z}^T\|_F^2$$

is precisely the set of $\tilde{Z}$ of the form $\tilde{Z} = Z_K\Lambda_K^{1/2}O_K$ for some top-K eigenvector-matrix Z_K of the GEP (M, I) and some orthogonal $O_K \in \mathcal{O}(K)$, and where Λ_K is a diagonal matrix of the top-K eigenvalues.

Proof. First note that when M is positive semi-definite the SVD coincides with the eigendecomposition.

Second note that taking $\tilde{Z} = Z_K\Lambda_K^{1/2}O_K$ attains the minimal value by the Eckhart–Young inequality, Theorem 1.1.

Next note that if $\tilde{Z}$ attains the minimal value then it must have $\tilde{Z}\tilde{Z}^T = Z_K\Lambda_KZ_K^T$ by the equality case of Eckhart–Young. Then by matrix square root Lemma 1.1 we must indeed have $\tilde{Z} = Z_K\Lambda_K^{1/2}O_K$ for some orthogonal O_K. □

Corollary 1.2 (Symmetric Eckhart–Young for square root matrix)**.** *Let $M \in \mathbb{R}^{d\times d}$ be symmetric with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d$ such that $\lambda_K > 0$. Then*

$$\underset{\tilde{Z}\in\mathbb{R}^{d\times K}}{\arg\min} \|M - \tilde{Z}\tilde{Z}^T\|_F^2$$

is precisely the set of $\tilde{Z}$ of the form $\tilde{Z} = Z_K\Lambda_K^{1/2}O_K$ for some top-K eigenvector-matrix Z_K of the GEP (M, I) and some orthogonal $O_K \in \mathcal{O}(K)$, and where Λ_K is a diagonal matrix of the top-K eigenvalues.

Proof. Let $\tilde{Z} \in \mathbb{R}^{d\times K}$. Because M is symmetric it has some eigen-decomposition; separate this into strictly positive and non-positive eigenvalues $M = M_+ + M_- =$

$Z_+\Lambda_+Z_+^T + Z_-\Lambda_-Z_-^T$, with rank d_+, d_- respectively. Let the corresponding projections be $P_+ = Z_+Z_+^T, P_- = Z_-Z_-^T$.

Now define $\tilde{Z}_+ = P_+\tilde{Z}, \tilde{Z}_- = P_-\tilde{Z}$. Then note by orthogonality of the projections we have for any matrix A that

$$\|A\|^2 = \|(P_+ + P_-)A(P_+ + P_-)\|^2 = \|P_+AP_+\|^2 + \|P_+AP_-\|^2 + \|P_-AP_+\|^2 + \|P_-AP_-\|^2$$

So we can expand out

$$\begin{aligned}\|M - \tilde{Z}\tilde{Z}^T\|^2 &= \|(P_+ + P_-)(M - \tilde{Z}\tilde{Z}^T)(P_+ + P_-)\|^2 \\ &= \underbrace{\|M_+ - \tilde{Z}_+\tilde{Z}_+^T\|^2}_{\geq \sum_{k=K+1}^{d_+}\lambda_k^2} + \underbrace{\|M_- - \tilde{Z}_-\tilde{Z}_-^T\|^2}_{\geq \|M_-\|^2} + \underbrace{\|\tilde{Z}_+\tilde{Z}_-^T\|^2}_{\geq 0} + \underbrace{\|\tilde{Z}_-\tilde{Z}_+^T\|^2}_{\geq 0} \geq \sum_{k=K+1}^{d}\lambda_k^2\end{aligned} \tag{B.1}$$

where the first inequality follows from the previous Corollary 1.1 and the second inequality is just from

$$\|M_- - \tilde{Z}_-\tilde{Z}_-^T\|^2 - \|M_-\|^2 = -2\,\mathrm{trace}\left(\tilde{Z}_-^T M_-\tilde{Z}_-\right) + \|\tilde{Z}_-\tilde{Z}_-^T\|^2 \geq 0$$

because M_- has negative eigenvalues.

Moreover equality in (B.1) requires the equality case of all the component inequalities; the first gives $\tilde{Z}_+ = Z_K\Lambda_K^{1/2}O_K$ for some Z_K, O_K as in the statement of Corollary 1.1, and the second that $\tilde{Z}_- = 0$; so indeed combining $\tilde{Z} = \tilde{Z}_+ + \tilde{Z}_-$ gives the result. □

1.4 GEP-EY Objective

Proposition 1.1 (GEP-EY-Objective)**.** *Consider the GEP (A, B) with A symmetric and B positive definite; suppose there are at least K strictly positive (generalized) eigenvalues. Then:*

$$\tilde{W} \in \underset{\tilde{W}\in\mathbb{R}^{d\times k}}{\arg\max}\ \mathrm{trace}\left\{2\left(\tilde{W}^TA\tilde{W}\right) - \left(\tilde{W}^TB\tilde{W}\right)\left(\tilde{W}^TB\tilde{W}\right)\right\}$$

if and only if $\tilde{W} = W_K\Lambda_K^{1/2}O_K$ for some top-K matrix W_K of the GEP and some orthogonal $O_K \in \mathcal{O}(k)$, where Λ_K is a diagonal matrix of the top-K eigenvalues.

Moreover, the maximum value is precisely $\sum_{k=1}^{K}\lambda_k^2$.

Proof. First recall that there is a bijection between eigenvectors w for the GEP

(A, B) and eigenvectors $z = B^{1/2}w$ for the GEP (M, I) where $M := B^{-1/2}AB^{-1/2}$ (e.g. see Chapman, Aguila, and Wells (2022)).

Now consider how the Eckhart–Young objective from Corollary 1.2 transforms under the bijection $Z = B^{1/2}W$.

We get

$$\begin{aligned}\|M - \tilde{Z}\tilde{Z}\|_F^2 &= \|B^{-1/2}AB^{-1/2} - B^{1/2}\tilde{W}\tilde{W}^T B^{1/2}\|_F^2 \\ &= \|B^{-1/2}AB^{-1/2}\|_F^2 - 2\,\mathrm{trace}\left(B^{-1/2}AB^{-1/2}B^{1/2}\tilde{W}\tilde{W}^T B^{1/2}\right) \\ &\quad + \mathrm{trace}\left(B^{1/2}\tilde{W}\tilde{W}^T B^{1/2}\, B^{1/2}\tilde{W}\tilde{W}^T B^{1/2}\right) \\ &= \|B^{-1/2}AB^{-1/2}\|_F^2 - \mathrm{trace}\left\{2\left(\tilde{W}^T A\tilde{W}\right) - \left(\tilde{W}^T B\tilde{W}\right)\left(\tilde{W}^T B\tilde{W}\right)\right\},\end{aligned}$$

where the first term is independent of $\tilde{W}$, so we can conclude by Corollary 1.2.

The moreover conclusion can follow from computing the objective at any maximiser of the form above. We note that

$$\begin{aligned}\tilde{W}^T A\tilde{W} &= O_K^T \Lambda_K^{1/2} W_K^T A W_K \Lambda_K O_K = O_K^T \Lambda_K^2 O_K \\ \tilde{W}^T B\tilde{W} &= O_K^T \Lambda_K^{1/2} W_K^T B W_K \Lambda_K O_K = O_K^T \Lambda_K O_K\end{aligned}$$

plugging into the objective gives

$$\mathrm{trace}\left(2\,(\tilde{W}^T A\tilde{W}) - (\tilde{W}^T B\tilde{W})^2\right) = \mathrm{trace}\left(2\,O_K^T\Lambda_K^2 O_K - O_K^T\Lambda_K^2 O_K\right) = \sum_{k=1}^{K}\lambda_k^2$$

because the trace of a symmetric matrix is equal to the sum of its eigenvalues. □

2 Tractable Optimization - no spurious local minima

First in Section 2.1 we prove that for general A, B our loss $\mathcal{L}_{\mathsf{EY}}(U)$ has no spurious local minima. Then in Section 2.2 we apply a result from Ge, Jin, and Zheng (2017). This application is somewhat crude, and we expect that a quantitative result with tighter constants could be obtained by adapting the argument of Section 2.1; we leave such analysis to future work.

2.1 Qualitative results

First we prove an auxillary result.

Lemma 2.1. *Let $M \in \mathbb{R}^{D\times D}$ be a symmetric matrix and let $U \in \mathbb{R}^{D\times K}$. Let*

$$\hat{\Gamma} := \underset{\Gamma\in\mathbb{R}^{K\times K}}{\arg\min} \|M - U\Gamma U^T\|_F^2$$

Then $U\hat{\Gamma}U^T = \mathcal{P}_U M \mathcal{P}_U$ and the minimum value is precisely

$$\|M\|_F^2 - \|\mathcal{P}_U M \mathcal{P}_U\|_F^2 \tag{B.2}$$

Moreover, if U has orthonormal columns then $\hat{\Gamma} = U^T M U$, and $\|\mathcal{P}_U M \mathcal{P}_U\|_F^2 = \|\hat{\Gamma}\|_F^2$

Proof. Simply complete the square to give

$$\begin{aligned}\|M - U\Gamma U^T\|_F^2 &= \operatorname{trace}(U^TU)\Gamma^T(U^TU)\Gamma - 2\operatorname{trace} D(U^TMU) + \|M\|_F^2 \\ &= \|(U^TU)^{1/2}\Gamma(U^TU)^{1/2} - (U^TU)^{-1/2}(U^TMU)(U^TU)^{-1/2}\|_F^2 + \|M\|_F^2 - \|\mathcal{P}_U M\mathcal{P}_U\|_F^2\end{aligned}$$

from which we can read off that the minimum is attained precisely when

$$\Gamma = (U^TU)^{-1}(U^TMU)(U^TU)^{-1}$$

and that the optimal value is precisely the value of Equation (B.2) as claimed. Finally, if U has orthonormal columns, $U^TU = I_K$ so Γ^* is of the form claimed, and the final equality comes from expanding out the trace form of the Frobenius norm. □

Lemma 2.2. *Let $M \in \mathbb{R}^{D\times D}$ be a symmetric matrix and $\mathcal{U}$ a subspace of $\mathbb{R}^D$ of dimension L. Then there exists an orthonormal basis $u_1, \dots, u_L$ for $\mathcal{U}$ such that*

$$u_L \perp Mu_l \quad \textit{for } l \in \{1, \dots, L-1\}$$

Proof. Consider the action of $\tilde{M} := \mathcal{P}_{\mathcal{U}} M \mathcal{P}_{\mathcal{U}}$ on $\mathcal{U}$. Then $\tilde{M}$ is symmetric matrix whose range is a subspace of $\mathcal{U}$ and so there exists an orthonormal set of eigenvectors $u_1, \dots, u_L$ that give a basis for $\mathcal{U}$ with corresponding eigenvalues $\tilde{\lambda}_1 \geq \cdots \geq \tilde{\lambda}_L$. Then we can read off

$$\langle u_L, Mu_l\rangle = \langle u_L, \tilde{M}u_l\rangle = \tilde{\lambda}_l\langle u_L, u_l\rangle = 0$$

as required. □

Proposition 2.1 (No spurious local minima)**.** *The (population) objective $\mathcal{L}^{EY}$ has no spurious local minima. That is, any matrix $\bar{W}$ that is a local minimum of $\mathcal{L}^{EY}$*

must in fact be a global minimum of the form described in Proposition 3.1.

Proof. We shall show that for any matrix W that is not a global optimum, there is a (continuous) path of solutions W_t with:

$$W_0 = W, \quad W_1 = \hat{W}, \quad W_t \to W \text{ as } t \to 0, \quad \text{and} \quad \mathcal{L}^{EY}(W_t) < \mathcal{L}^{EY}(W) \, \forall t > 0$$

As in the proof of Proposition 3.1 we first reduce to the $B = I$ setting by defining $Z := B^{1/2}W$ and $M = B^{-1/2}AB^{-1/2}$. Let the eigendecomposition of M be $M = V^* D^* {V^*}^\top$. Define the loss

$$l(Z) := \|M - ZZ^\top\|_F^2$$

It is now sufficient to show that: for any matrix $Z \in \mathbb{R}^{D \times K}$ that is not of the form $V_K^* D_K^* O_K$ where V_K^* is a matrix whose columns are a set of top-K eigenvectors for M, and $O_K \in \mathbb{R}^{K \times K}$ is some arbitrary orthogonal matrix cannot be a local minimum.

For notational simplicity we will assume that the $\lambda_K(M) > \lambda_{K+1}(M)$ from now on, such that V_K^* can be made well-defined[1].

Now, take such a Z and suppose, for contradiction that it is a local minimum. We will construct a continuous path of matrices $Z(t) : t \in [0, 1]$ with $Z(0) = Z$ and $l\left(Z(t)\right) < l(Z) \, \forall t > 0$.

Then by our assumption on the form of Z, we have

$$\mathcal{V}_K := \operatorname{span}\{Z\} \neq \operatorname{span}\{V_K^*\} =: \mathcal{V}_K^*$$

Now comes the clever part of the proof. Define $\kappa_\cap = \dim \operatorname{span}\{\mathcal{V}_K \cap \mathcal{V}_K^*\}$. Then pick orthonormal bases

- $u_1, \ldots, u_{\kappa_\cap}$ for $\mathcal{V}_K \cap \mathcal{V}_K^*$
- $u_{\kappa_\cap + 1}, \ldots, u_K$ for $\mathcal{V}_K \cap \mathcal{V}_K^*$ such that $u_K \perp M u_k$ for all $k = \kappa_\cap + 1, \ldots, K-1$ by Lemma 2.2
- $u^*_{\kappa_\cap + 1}, \ldots, u^*_K$ for $\mathcal{V}_K^* \cap \mathcal{V}_K$

Let $U_K = \begin{pmatrix} & & \\ u_1 & \ldots & u_K \\ & & \end{pmatrix}$. Then by Lemma 2.1, for Z to be a local minimum

[1] with symmetry breaking for earlier repeated eigenvalues if required.

we must have

$$ZZ^T = U_K(U_K^T M U_K)U_K^T$$

Moreover the objective value must therefore be

$$l(Z) = \|M\|_F^2 - \|U_K^T M U_K\|_F^2 \tag{B.3}$$

We now make the observation that the second term is the 'signal of M captured by the subspace of U_K'. So aligning U_K with higher-eigenvalue subspaces of M should increase this amount of signal captured and decrease this loss.

We now construct a path $U_K(t)$ which captures this intuition.

Let $u_K(t) = \cos(t)u_K + \sin(t)u_K^*$. Then let $U_K(t)$ have columns $u_1, \ldots, u_{K-1}, u_K(t)$. By construction this is still an orthonormal set of basis vectors, so $U_K(t)^T U_K = I_K$. Let $\Gamma(t) = U_K(t)^T M U_K(t)$.

We are finally ready to construct the path $Z(t)$. Because U_K is a basis for the column space of Z, and Z is assumed to be a local optimum, we must have

$$ZZ^T = U_K \Gamma(0) U_K^T$$

by Lemma 2.1. So $Z = U_K \Gamma^{1/2} O_K$ for some orthogonal matrix $O_K \in \mathbb{R}^{K \times K}$ where $\Gamma^{1/2}$ is the unique positive semi-definite square root of Γ. So define

$$Z(t) = U_K(t)\Gamma(t)^{1/2} O_K$$

where again $\Gamma(t)^{1/2}$ is the unique positive semi-definite square root and therefore both $U_k(t)$ and $\Gamma(t)^{1/2}$ are continuous functions of t and therefore so is Z.

Then

$$l(Z(t)) = \|M\|_F^2 - \|U_K(t)^T M U_K(t)\|_F^2 \tag{B.4}$$

So it is sufficient to show that $\|U_K(t)^T M U_K(t)\|_F^2 > \|U_K^T M U_K\|_F^2$ for $t \in [0, \pi/2]$.

Indeed, we can compute

$$\begin{aligned}\|U_K(t)^T M U_K(t)\|_F^2 - \|U_K^T M U_K\|_F^2 &= \left(u_K(t)^T M u_K(t)\right)^2 - \left(u_K^T M u_K\right)^2 \\ &\quad + 2\sum_{k=1}^{K-1}\left\{\left(u_K(t)^T M u_k\right)^2 - \left(u_K^T M u_k\right)^2\right\} \\ &\geq \left(u_K(t)^T M u_K(t)\right)^2 - \left(u_K^T M u_K\right)^2\end{aligned}$$

because $u_K^T M u_k = 0$ for $k = 1, \ldots, K-1$ by construction. Finally we have

$$\begin{aligned}u_K(t)^T M u_K(t) &= \sin^2(t)\langle u_K^*, M u_K^*\rangle + 2\sin(t)\cos(t)\langle u_K, M u_K^*\rangle + \cos^2(t)\langle u_K, M u_K\rangle \\ &= \sin^2(t)\langle u_K^*, M u_K^*\rangle + \cos^2(t)\langle u_K, M u_K\rangle \\ &> u_K^T M u_K\end{aligned}$$

Here we used that $\langle u_K^*, M u_K^*\rangle \geq \lambda_K > \langle u_K, M u_K\rangle$ and that the middle term vanishes because $M u_K^* \in \mathcal{U}_K^*$ and is therefore orthogonal to u_K.

□

2.2 Quantitative results

To use the results from Ge, Jin, and Zheng (2017) we need to introduce their definition of a (θ, γ, ζ)-strict saddle.

Definition 2.1. *We say function $l(\cdot)$ is a (θ, γ, ζ)-**strict saddle** if for any x, at least one of the following holds:*

1. $\|\nabla l(x)\| \geq \theta$
2. $\lambda_{min}(\nabla^2 l(x)) \leq -\gamma$
3. *x is ζ-close to $\mathcal{X}^*$ - the set of local minima.*

.

We can now state restate Lemma 13 from Ge, Jin, and Zheng (2017) in our notation; this was used in their analysis of robust PCA, and directly applies to our PCA-type formulation.

Lemma 2.3 (Strict saddle for PCA)**.** *Let $M \in \mathbb{R}^{D\times D}$ be a symmetric PSD matrix, and define the matrix factorization objective over $Z \in \mathbb{R}^{D\times K}$*

$$l(Z) = \|M - ZZ^\top\|^2$$

Assume that $\lambda_K^* := \lambda_K(M) \geq 15\lambda_{K+1}(M)$. Then

1. *all local minima satisfy $ZZ^T = \mathcal{P}_K(M)$ - the best rank-K approximation to M*
2. *the objective l(Z) is $(\epsilon, \Omega(\lambda_K^*), \mathcal{O}(\epsilon/\lambda_K^*)$-strict saddle.*

However, we do not want to show a strict saddle of l but of $\mathcal{L}_{\mathsf{EY}} : U \mapsto l(B^{1/2}U)$. Provided that B has strictly positive minimum and bounded maximum eigenvalues this implies that $\mathcal{L}_{\mathsf{EY}}$ is also strict saddle, as we now make precise.

Lemma 2.4 (Change of variables for strict saddle conditions)**.** *Suppose that l is (θ, γ, ζ)-strict saddle and let $L : U \mapsto l(B^{1/2}U)$ for B with minimal and maximal eigenvalues $\sigma_{min}, \sigma_{max}$ respectively.*

Then L is $\left(\sigma_{max}^{1/2}\theta,\ \sigma_{min}\gamma,\ \sigma_{max}^{1/2}\zeta\right)$-strict saddle.

Proof. Write $g(U) = B^{1/2}U$. Then $L = l \circ g$, so by the chain rule:

$$D_U L = D_{B^{1/2}U} l \circ D_U g \ :\ \delta U \mapsto \langle \nabla l(B^{1/2}U), B^{1/2}\delta U\rangle = \langle B^{1/2}\nabla l(B^{1/2}U), \delta U\rangle$$

Therefore

$$\|\nabla L(U)\| = \|B^{1/2}\nabla l(B^{1/2}U)\| \geq \sigma_{\mathsf{min}}^{1/2}\|l(B^{1/2}U)\|$$

By a further application of the chain rule we have

$$D_U^2 L \ :\ \delta U, \delta U \mapsto D^2_{B^{1/2}U} l(B^{1/2}\delta U, B^{1/2}\delta U)$$

Suppose $\lambda_{\mathsf{min}}(\nabla^2 l(Z)) \leq -\gamma$ then by the variational characterization of eigenvalues, there exists some δZ such that $\langle \delta Z, \nabla^2 l(Z)\delta Z\rangle \leq -\gamma\|\delta Z\|^2$. Then taking $\delta U = B^{-1/2}\delta Z$ gives

$$\begin{aligned}
\langle \delta U, \nabla^2 L(U)\delta U\rangle &= \langle B^{1/2}\delta U, \nabla^2 l(B^{1/2}U)B^{1/2}\delta U\rangle \\
&= \langle \delta Z, \nabla^2 l(Z)\delta Z\rangle \\
&\leq -\gamma\|\delta Z\|^2 \\
&\leq -\gamma\sigma_{\mathsf{min}}\|\delta U\|^2
\end{aligned}$$

Thirdly, suppose that $\|B^{1/2}U - Z^*\| \leq \zeta$ for some local optimum Z^* of l. Then since B is invertible, $U^* := B^{-1/2}Z^*$ is a local optimum of L. In addition:

$$\|U - U^*\| = \|B^{1/2}(U - U^*)\| \leq \sigma_{\mathsf{max}}^{1/2}\|B^{1/2}U - Z^*\| \leq \zeta$$

Finally, consider some arbitrary point U_0. Let $Z_0 = B^{1/2}U_0$. Then by the strict saddle property for l one of the following must hold:

1. $\|\nabla l(Z_0)\| \geq \theta \quad \implies \quad \|\nabla L(U_0)\| \geq \sigma_{\text{min}}^{1/2}\theta$
2. $\lambda_{\text{min}}(\nabla^2 l(Z_0)) \leq -\gamma \quad \implies \quad \lambda_{\text{min}}(\nabla^2 L(U_0) \leq -\sigma_{\text{min}}\gamma$
3. Z_0 is ζ-close to a local-minimum Z^*, which implies that U_0 is $(\sigma_{\text{max}}^{1/2}\zeta)$-close to a local minimum $B^{-1/2}Z^*$ of L.

□

By combining Lemma 2.3 with Lemma 2.4, we can conclude that our objective does indeed satisfy a (quantitative) strict saddle property. This is sufficient to show that certain local search algorithms will converge in polynomial time Ge, Jin, and Zheng, 2017.

3 Fast updates for (Multi-view) Stochastic CCA (and PLS)

3.1 Back-propagation for empirical covariances

To help us analyse the full details of back-propagation in the linear case, we first prove a lemma regarding the gradients of the empirical covariance operator.

Lemma 3.1 (Back-prop for empirical covariance)**.** *Let* $e \in \mathbb{R}^M, f \in \mathbb{R}^M$*. Then* $\widehat{\mathrm{Cov}}(e, f)$ *and*

$$\frac{\partial \widehat{\mathrm{Cov}}(e, f)}{\partial e}$$

can both be computed in $\mathcal{O}(M)$ *time.*

Proof. Let $1_M \in \mathbb{R}^M$ be a vector of ones and $\mathcal{P}^{\perp}_{1_M} = I_M - \frac{1}{M} 1^T_M 1_M$ be the projection away from this vector, then we can write $\bar{e} = \mathcal{P}^{\perp}_{1_M} e, \bar{f} = \mathcal{P}^{\perp}_{1_M} f$. Moreover, exploiting the identity-plus-low-rank structure of $\mathcal{P}^{\perp}_{1_M}$ allows us to compute these quantities in $\mathcal{O}(M)$ time.

Then by definition

$$\widehat{\mathrm{Cov}}(e, f) = \frac{1}{M-1} \bar{e}^T \bar{f}$$

which is again computable in $\mathcal{O}(M)$ time.

For the backward pass, first note that

$$\frac{\partial \bar{e}}{\partial e} \,:\, \delta e \mapsto \mathcal{P}^{\perp}_{1_M} \delta e$$

So the derivative with respect to e is

$$\frac{\partial \widehat{\mathrm{Cov}}(e, f)}{\partial e} = \frac{1}{M-1} \frac{\partial \bar{e}^T \bar{f}}{\partial e} = \frac{1}{M-1} \left(\frac{\partial \bar{e}}{\partial e} \bar{f} \right) = \frac{1}{M-1} \mathcal{P}^{\perp}_{1_M} \bar{f} = \frac{1}{M-1} \bar{f}$$

because $\bar{f}$ is independent of e, and already mean-centred. So all that remains is element-wise division, which again costs $\mathcal{O}(M)$ time. □

Forward Pass

1. **Compute the transformed variables $\mathbf{Z}$:**

$$\mathbf{Z}^{(i)} = U^{(i)}\mathbf{X}^{(i)}, \tag{B.5}$$

   with a complexity of $\mathcal{O}(MKD)$.

2. **Compute $\operatorname{trace}\hat{C}(\theta)[\mathbf{Z}]$:** the diagonal elements of $\hat{C}$ are simply

$$\hat{C}_{kk} = \sum_{i \neq j} \widehat{\operatorname{Cov}}(\mathbf{Z}_k^{(i)}, \mathbf{Z}_k^{(j)})$$

   which each summand can be computed in $\mathcal{O}(M)$ time, so summing over i, j, k gives total complexity of $\mathcal{O}(I^2KM)$.

3. **Compute $\hat{V}(\theta)[\mathbf{Z}]$:** For $\hat{V}_\alpha[\mathbf{Z}]$:

$$\hat{V}_\alpha(\theta)[\mathbf{Z}] = \sum_i \alpha_i {U^{(i)}}^T U^{(i)} + (1-\alpha_i)\widehat{\operatorname{Var}}(\mathbf{Z}^{(i)}),$$

   each ${U^{(i)}}^T U^{(i)}$ can be computed with a complexity of $\mathcal{O}\left(D_i K^2\right)$ and the total cost of evaluating all of these is $\mathcal{O}\left(K^2 D\right)$. Each summand in the second term costs $\mathcal{O}\left(MK^2\right)$ by Lemma 3.1 so evaluating the full second term costs $\mathcal{O}\left(IMK^2\right)$.

4. **Evaluate $\hat{\mathcal{L}}_{\mathsf{EY}}[\mathbf{Z}, \mathbf{Z}']$:**

$$\hat{\mathcal{L}}_{\mathsf{EY}}[\mathbf{Z}, \mathbf{Z}'] = -2 \operatorname{trace}\hat{C}[\mathbf{Z}] + \langle \hat{V}_\alpha[\mathbf{Z}], \hat{V}_\alpha[\mathbf{Z}'] \rangle_F. \tag{B.6}$$

   The dominant complexity here is the $\mathcal{O}(K^2)$ cost of computing the Frobenius inner product.

Backward Pass

1. **Gradient with respect to $\mathbf{Z}^{(i)}$:** Using the chain rule, the gradient will flow back from the final computed value, $\hat{\mathcal{L}}_{\mathsf{EY}}[\mathbf{Z}, \mathbf{Z}']$, through the operations that produced it.

2. **Gradient of $\operatorname{trace}\hat{C}(\theta)[\mathbf{Z}]$ with respect to $\mathbf{Z}_k^{(i)}$:** Is precisely

$$\frac{\partial \hat{C}_{kk}}{\partial \mathbf{Z}_k^{(i)}} = \frac{2}{M-1}\sum_{j\neq i}\bar{\mathbf{Z}}_k^{(j)},$$

where $\bar{\mathbf{Z}}_k^{(j)} = \mathcal{P}_{1_M}^{\perp}\bar{\mathbf{Z}}_k^{(j)}$, from Lemma 3.1 and so can be computed in $\mathcal{O}(IM)$ time.

3. **Gradients of $\langle \hat{V}_\alpha[\mathbf{Z}], \hat{V}_\alpha[\mathbf{Z}']\rangle_F$ with respect to $\mathbf{Z}_k^{(i)}$:** By applying Lemma 3.1, the gradient of the empirical variance term is

$$\frac{\partial \widehat{\operatorname{Var}}(\mathbf{Z}^{(i)})_{l,l'}}{\partial \mathbf{Z}_k^{(i)}} = \begin{cases} \frac{2}{M-1}\mathbf{Z}_k^{(i)} & \text{if } l = l' = k \\ \frac{1}{M-1}\mathbf{Z}_l^{(i)} & \text{if } l \neq l' = k \\ 0 & \text{otherwise.} \end{cases}$$

and so

$$\begin{aligned}\frac{\partial \langle \hat{V}_\alpha[\mathbf{Z}], \hat{V}_\alpha[\mathbf{Z}']\rangle_F}{\partial \mathbf{Z}_k^{(i)}} &= \frac{(1-\alpha_i)}{M-1}\left(2\hat{V}_\alpha[\mathbf{Z}']_{kk}\mathbf{Z}_k^{(i)} + \sum_l (\hat{V}_\alpha[\mathbf{Z}']_{lk}\mathbf{Z}_l^{(i)} + \hat{V}_\alpha[\mathbf{Z}']_{kl}\mathbf{Z}_k^{(i)})\right) \\ &= \frac{2(1-\alpha_i)}{M-1}\sum_{l=1}^{K}\hat{V}_\alpha[\mathbf{Z}']_{lk}\mathbf{Z}_l^{(i)}\end{aligned}$$

this can be computed in $\mathcal{O}(MK)$ time.

4. **Gradients of $\hat{\mathcal{L}}_{\mathsf{EY}}[\mathbf{Z}, \mathbf{Z}']$ with respect to $\mathbf{Z}_k^{(i)}$:** can therefore be computed for a given $\mathbf{Z}_k^{(i)}$ in $\mathcal{O}\left(M(K+I)\right)$ time and so, adding up over all i, k gives total $\mathcal{O}\left(IM(K+I)\right)$ time.

5. **Gradients of $\langle \hat{V}_\alpha[\mathbf{Z}], \hat{V}_\alpha[\mathbf{Z}']\rangle_F$ with respect to $U_k^{(i)}$:** is similarly

$$\frac{2\alpha_i}{M-1}\sum_{l=1}^{K}(\hat{V}_\alpha[\mathbf{Z}]_{lk} + \hat{V}_\alpha[\mathbf{Z}']_{lk})U_l^{(i)}$$

so can be computed in $\mathcal{O}\left(D_i K\right)$ time.

6. **Finally compute gradients with respect to $U_k^{(i)}$:** simply have $Z_k^{(i)} =$

${U_k^{(i)}}^\top \mathbf{X}^{(i)}$ so the final gradients are

$$\frac{\partial \hat{\mathcal{L}}_{\mathsf{EY}}}{\partial U_k^{(i)}} = \left(\frac{\partial \hat{\mathcal{L}}_{\mathsf{EY}}}{\partial \mathbf{Z}_k^{(i)}}\right)^\top \mathbf{X}^{(i)} + \frac{\partial \langle \hat{V}_\alpha[\mathbf{Z}], \hat{V}_\alpha[\mathbf{Z}'] \rangle_F}{\partial U_k^{(i)}} \tag{B.7}$$

so the dominant cost is the $\mathcal{O}(MD_i)$ multiplication.

Since $D \gg K, M$, the dominant cost each final gradient is $\mathcal{O}(MD_i)$. Summing up over i, k gives total cost $\mathcal{O}(KM \sum D_i) = \mathcal{O}(KMD)$, as claimed.

Appendix C

Proofs and Additional Results for Chapter VI

1 Eckhart-Young loss recovers Deep CCA

Lemma 3.1. *[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 Andrew et al. (2013) and Somandepalli et al. (2019) as defined in Equation (VI.1).

Proof. Write $f^{(i)}(X^{(i)};\theta^{(i)}) = U^{(i)\,T} g^{(i)}(X^{(i)};\phi^{(i)})$ where the $U^{(i)}$ are matrices parameterising the final layer and $g^{(i)}$ defines the representations in the penultimate layer.

Because $\hat{\theta}$ is a local minimum of $\mathcal{L}_{\mathsf{EY}}(\theta)$ we must have $\hat{U}$ a local minimum of the map $l : U \mapsto \mathcal{L}_{\mathsf{EY}}((U, \hat{\phi}))$. Writing $\hat{Y} = g(X;\hat{\phi})$ for the corresponding penultimate-layer representations we get

$$\begin{aligned} l(U) := \mathcal{L}_{\mathsf{EY}}((U,\hat{\phi})) &= -2\,\mathrm{trace}\,\big(\sum_{i\neq j}\mathrm{Cov}(U^{(i)\,T}\hat{Y}^{(i)}, U^{(j)\,T}\hat{Y}^{(j)})\big) + \big\|\sum_i \mathrm{Var}(U^{(i)\,T}\hat{Y}^{(i)})\big\|_F^2 \\ &= -2\,\mathrm{trace}\,\Big(U^T A(\hat{Y})U\Big) + \|U^T B(\hat{Y})U\|_F^2 \end{aligned}$$

where $A(\hat{Y}), B(\hat{Y})$ are as in **??** with X replaced by $\hat{Y}$. This is precisely our Eckhart-Young loss for linear CCA on the $\hat{Y}$. So by Proposition 3.2, $\hat{U}$ must also be a global minimum of $l(U)$ and then by Proposition 3.1 the optimal value is precisely $-\|\mathrm{MCCA}_K(\hat{Y})\|_2^2$.

This in turn is equal to $-\|\mathrm{MCCA}_K(\hat{Z})\|_2^2$ by a simple sandwiching argument. Indeed, by Proposition 3.1 $\min_V \mathcal{L}_{\mathsf{EY}}((V^{(i)\,T}X^{(i)})_i) = -\|\mathrm{MCCA}_K(\hat{Z})\|_2^2$. Then we can chain inequalities

$$\begin{aligned}-\|\mathrm{MCCA}_K(\hat{Y})\|_2^2 = \mathcal{L}_{\mathsf{EY}}(\hat{Z}) &\geq \min_V \mathcal{L}_{\mathsf{EY}}((V^{(i)\,T}X^{(i)})_i) \\ &\geq \min_U \mathcal{L}_{\mathsf{EY}}((U^{(i)\,T}\hat{Y}^{(i)})_i) = -\|\mathrm{MCCA}_K(\hat{Y})\|_2^2\end{aligned}$$

to conclude. □

1.1 Interlacing results

First we state a standard result from matrix analysis. This is simply Theorem 2.1 from Haemers (1995), but with notation changed to match our context. We therefore omit the (straightforward) proof.

Lemma 1.1. *Let $Z \in \mathbb{R}^{D\times K}$ such that $Z^TZ = I_K$ and let $M \in \mathbb{R}^{D\times D}$ be symmetric with an orthonormal set of eigenvectors $v_1, \ldots, v_D$ with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_D$. Define $C = Z^TMZ$, and let C have eigenvalues $\mu_1 \geq \cdots \geq \mu_K$ with respective eigenvectors $y_1 \ldots y_K$.*

Then

- *$\mu_k \leq \lambda_k$ for $k = 1, \ldots, K$.*
- *if $\mu_k = \lambda_k$ for some k then C has a μ_k-eigenvector y such that Zy is a μ_k-eigenvector of M.*
- *if $\mu_k = \lambda_k$ for $k = 1, \ldots, K$ then Zy_k is a μ_k-eigenvector of M for $k = 1, \ldots, K$.*

This immediately gives us a related result for generalized eigenvalues.

Corollary 1.1 (Generalized Eigenvalue Interlacing)**.** *Consider the GEP (A, B) where $A \in \mathbb{R}^{D\times D}$ is symmetric and $B \in \mathbb{R}^{D\times D}$ symmetric positive definite; let these have B-orthonormal generalized eigenvectors $u_1, \ldots, u_D$ with eigenvalues $\lambda_1, \ldots, \lambda_D$.*

Let $U \in \mathbb{R}^{D\times K}$ such that $U^TBU = I_K$, define $C = U^TAU$, and let C have eigenvalues $\mu_1 \geq \cdots \geq \mu_K$ with respective eigenvectors $y_1 \ldots y_K$.

Then

- $\mu_k \leq \lambda_k$ *for* $k = 1, \ldots, K$.
- *if* $\mu_k = \lambda_k$ *for some* k *then* (C, V) *has a* μ_k*-generalised-eigenvector* y *such that* Uy *is a* μ_k*-generalised-eigenvector of* (A, B).
- *if* $\mu_k = \lambda_k$ *for* $k = 1, \ldots, K$ *then* Uy_k *is a* μ_k*-generalised-eigenvector of* (A, B) *for* $k = 1, \ldots, K$.

Proof. As in previous appendices, we convert from the GEP (A, B) to an eigenvalue problem for $M := B^{-1/2}AB^{-1/2}$ by defining $Z = B^{-1/2}U$, and $v_d = B^{1/2}u_d$.

We now check that the conditions and conclusions of Lemma 1.1 biject with the conditions and conclusions of this present lemma.

Indeed $(u_d)_d$ are B-orthonormal gevectors of (A, B) if and only if $(v_d)_d$ are orthonormal evectors of M; the matrices C and then coincide and so so does its eigenvectors and eigenvalues.

This proves the result. □

We can now apply this to the Multi-view CCA problem, generalising the two-view case.

Lemma 1.2 (Interlacing for MCCA)**.** *Let* $(X^{(i)})_{i=1}^I$ *be random vectors taking values in* $\mathbb{R}^{D_i}$ *respectively, as in Section 2. Take arbitrary full-rank weight matrices* $U^{(i)} \in \mathbb{R}^{D_i \times K}$ *for* $i \in \{1, \ldots, I\}$ *and define the corresponding transformed variables* $Z^{(i)} = \langle U^{(i)}, X^{(i)} \rangle$*. Then we have the element-wise inequalities*

$$\mathrm{MCCA}_K(Z^{(i)}, \ldots, Z^{(I)}) \leq \mathrm{MCCA}_K(X^{(1)}, \ldots, X^{(I)}) \tag{C.1}$$

Moreover simultaneous equality in each component holds if and only if there exist matrices $Y^{(i)} \in \mathbb{R}^{K \times K}$ *for* $i \in [I]$ *such that the* $(U^{(i)}Y^{(i)})_{i=1}^I$ *are a set of top-*K *weights for the MCCA problem.*

Proof. Let the matrices A, B be those from the MCCA GEP in **??** defined by the input variables X. By definition, $\mathrm{MCCA}_K(X^{(1)}, \ldots, X^{(I)})$ is precisely the vector of the top-K such generalised eigenvalues.

Then the corresponding matrices defining the GEP for Z are block matrices $\bar{A}, \bar{B}$ defined by the blocks

$$\begin{aligned} \bar{A}^{(ij)} &= \mathrm{Cov}(Z^{(i)}, Z^{(j)}) = {U^{(i)}}^\top \mathrm{Cov}(X^{(i)}, X^{(j)})\, U^{(j)} \\ \bar{B}^{(ii)} &= \mathrm{Var}(Z^{(i)}) = {U^{(i)}}^\top \mathrm{Var}(X^{(i)})\, U^{(i)} \end{aligned} \tag{C.2}$$

Now define the $D \times (KI)$ block diagonal matrix $\tilde{U}$ to have diagonal blocks $U^{(i)}$. Then the definition from Equation (C.2) is equivalent to the block-matrix equations $\bar{A} = \bar{U}^T A \bar{U}$, $\bar{B} = \bar{U}^T B \bar{U}$, both in $\mathbb{R}^{(KI)\times(KI)}$. Finally, we define a normalised version $\hat{U} = \bar{U}\bar{B}^{-1/2}$ (possible because B positive definite and $\bar{U}$ of full rank).

We can now apply the eigenvalue interlacing result of Corollary 1.1 to the GEP (A, B) and B-orthonormal matrix $\hat{U} \in \mathbb{R}^{D\times IK}$. Let the matrix $\bar{B}^{-1/2}\bar{A}\bar{B}^{-1/2} = \hat{U}^T A \hat{U}$ have top-K eigenvalues $\rho_1 \geq \cdots \geq \rho_K$ with respective eigenvectors $y_1, \ldots, y_K$. Then the $(\rho_k)_{k=1}^K$ are precisely the first K successive multi-view correlations between the $Z^{(i)}$. As before, the first K successive multi-view correlations ρ_k^* between the $X^{(i)}$ are precisely the first K generalised eigenvalues of the GEP (A, B). We therefore we have the element-wise inequalities $\rho_k \leq \rho_k^*$ for each $k = 1, \ldots, K$.

Moreover, equality for each of the top-K multi-view correlations implies that $\hat{U}y_k$ is a generalised-eigenvector of the original GEP (A, B) for $k = 1, \ldots K$ (still by Corollary 1.1). Letting $Y^{(i)} = \begin{pmatrix} y_1^{(i)} & \cdots & y_K^{(i)} \end{pmatrix}$ then gives the equality case statement.

□

References

Adams, Rick A. et al. (2024). “Voxel-wise multivariate analysis of brain-psychosocial associations in adolescents reveals six latent dimensions of cognition and psychopathology”. In: *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*. ISSN: 2451-9022. DOI: https://doi.org/10.1016/j.bpsc.2024.03.006. URL: https://www.sciencedirect.com/science/article/pii/S2451902224000855.

Akaho, Shotaro (2006). “A kernel method for canonical correlation analysis”. In: *arXiv preprint cs/0609071*.

Ali, Alnur, Edgar Dobriban, and Ryan Tibshirani (2020). “The implicit regularization of stochastic gradient flow for least squares”. In: *International conference on machine learning*. PMLR, pp. 233–244.

Alpert, Mark I and Robert A Peterson (1972). “On the interpretation of canonical analysis”. In: *Journal of marketing Research* 9.2, pp. 187–192.

Altmann, Andre et al. (2023). “Tackling the dimensions in imaging genetics with CLUB-PLS”. In: *arXiv preprint arXiv:2309.07352*.

Amari, Shun-ichi (1993). “Backpropagation and stochastic gradient descent method”. In: *Neurocomputing* 5.4-5, pp. 185–196.

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

Arora, Raman, Andrew Cotter, et al. (2012). “Stochastic optimization for PCA and PLS”. In: *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, pp. 861–868.

Arora, Raman, Poorya Mianjy, and Teodor Marinov (2016). “Stochastic optimization for multiview representation learning using partial least squares”. In: *International Conference on Machine Learning*. PMLR, pp. 1786–1794.

Ashburner, John et al. (2014). “SPM12 manual”. In: *Wellcome Trust Centre for Neuroimaging, London, UK* 2464.4.

Babuschkin, Igor et al. (2020). *The DeepMind JAX Ecosystem*. URL: http://github.com/deepmind.

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.

Balakrishnama, Suresh and Aravind Ganapathiraju (1998). "Linear discriminant analysis-a brief tutorial". In: *Institute for Signal and information Processing* 18.1998, pp. 1–8.

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, Mark Ibrahim, et al. (2023). "A Cookbook of Self-Supervised Learning". In: *arXiv preprint arXiv:2304.12210*.

Balestriero, Randall and Yann LeCun (2022). "Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods". In: *arXiv preprint arXiv:2205.11508*.

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

Benton, Adrian et al. (2017). "Deep generalized canonical correlation analysis". In: *arXiv preprint arXiv:1702.02519*.

Bhatia, Kush et al. (2018). "Gen-oja: Simple & efficient algorithm for streaming generalized eigenvector computation". In: *Advances in neural information processing systems* 31.

Bhatia, Rajendra (1997). *Matrix Analysis*. Vol. 169. Graduate Texts in Mathematics. New York, NY: Springer. ISBN: 978-1-4612-6857-4 978-1-4612-0653-8. DOI: 10.1007/978-1-4612-0653-8. URL: http://link.springer.com/10.1007/978-1-4612-0653-8 (visited on 03/21/2023).

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

Bilenko, Natalia Y and Jack L Gallant (2016). "Pyrcca: regularized kernel canonical correlation analysis in python and its applications to neuroimaging". In: *Frontiers in neuroinformatics* 10, p. 49. DOI: 10.3389/fninf.2016.00049.

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.

Bogdan, Ryan et al. (2017). "Imaging genetics and genomics in psychiatry: a critical review of progress and potential". In: *Biological psychiatry* 82.3, pp. 165–175.

Borga, Magnus (1998). “Learning Multidimensional Signal Processing”. eng. Publisher: Linköping University Electronic Press. PhD thesis. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-54341 (visited on 10/13/2022).

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.

Carlsson, Marcus (Mar. 2021). “von Neumann’s trace inequality for Hilbert–Schmidt operators”. en. In: *Expositiones Mathematicae* 39.1, pp. 149–157. ISSN: 0723-0869. DOI: 10.1016/j.exmath.2020.05.001. URL: https://www.sciencedirect.com/science/article/pii/S0723086920300220 (visited on 01/04/2023).

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 and Hao-Ting Wang (2021). “CCA-Zoo: A collection of Regularized, Deep Learning based, Kernel, and Probabilistic CCA methods in a scikit-learn style framework”. In: *Journal of Open Source Software* 6.68, p. 3823.

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.

Chapman, James, Lennie Wells, and Ana Lawry Aguila (2024). *Unconstrained Stochastic CCA: Unifying Multiview and Self-Supervised Learning*.

Chaudhari, Pratik and Stefano Soatto (2018). "Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks". In: *2018 Information Theory and Applications Workshop (ITA)*. IEEE, pp. 1–10.

Chen, Man-Sheng et al. (2022). "Representation learning in multi-view clustering: A literature review". In: *Data Science and Engineering* 7.3, pp. 225–241.

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.

Cuingnet, Rémi et al. (2012). "Spatial and anatomical regularization of SVM: a general framework for neuroimaging data". In: *IEEE transactions on pattern analysis and machine intelligence* 35.3, pp. 682–696.

Curth, Alicia, Alan Jeffares, and Mihaela van der Schaar (2023). "A U-turn on Double Descent: Rethinking Parameter Counting in Statistical Learning". In: *arXiv preprint arXiv:2310.18988*.

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.

Deng, Lingli et al. (2021). “Sparse PLS-based method for overlapping metabolite set enrichment analysis”. In: *Journal of proteome research* 20.6, pp. 3204–3213.

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.

Ermolov, Aleksandr et al. (2021). “Whitening for self-supervised representation learning”. In: *International Conference on Machine Learning*. PMLR, pp. 3015–3024.

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.

Falcon, William A (2019). “Pytorch lightning”. In: *GitHub* 3.

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.

Folstein, Marshal F, Susan E Folstein, and Paul R McHugh (1975). ““Mini-mental state”: a practical method for grading the cognitive state of patients for the clinician”. In: *Journal of psychiatric research* 12.3, pp. 189–198.

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, Furong Huang, et al. (2015). *Escaping From Saddle Points — Online Stochastic Gradient for Tensor Decomposition*. arXiv: 1503.02101 [cs.LG].

Ge, Rong, Chi Jin, Praneeth Netrapalli, et al. (2016). “Efficient algorithms for large-scale generalized eigenvector computation and canonical correlation analysis”. In: *International Conference on Machine Learning*. PMLR, pp. 2741–2750.

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.

Genon, Sarah, Simon B Eickhoff, and Shahrzad Kharabian (2022). “Linking interindividual variability in brain structure to behaviour”. In: *Nature Reviews Neuroscience* 23.5, pp. 307–318.

Ghojogh, Benyamin, Fakhri Karray, and Mark Crowley (2019). “Eigenvalue and generalized eigenvalue problems: Tutorial”. In: *arXiv preprint arXiv:1903.11240*.

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.

Gönen, Mehmet and Ethem Alpaydın (2011). “Multiple kernel learning algorithms”. In: *The Journal of Machine Learning Research* 12, pp. 2211–2268.

Goyal, Priya et al. (2019). “Scaling and benchmarking self-supervised visual representation learning”. In: *Proceedings of the ieee/cvf International Conference on computer vision*, pp. 6391–6400.

Greenacre, Michael et al. (2022). “Principal component analysis”. In: *Nature Reviews Methods Primers* 2.1, p. 100.

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

Gu, Fei and Hao Wu (2018). “Simultaneous canonical correlation analysis with invariant canonical loadings”. In: *Behaviormetrika* 45, pp. 111–132.

Guo, Wenzhong, Jianwen Wang, and Shiping Wang (2019). “Deep multimodal representation learning: A survey”. In: *Ieee Access* 7, pp. 63373–63394.

Haemers, Willem H. (Sept. 1995). “Interlacing eigenvalues and graphs”. en. In: *Linear Algebra and its Applications* 226-228, pp. 593–616. ISSN: 00243795. DOI: 10.1016/0024-3795(95)00199-2. URL: https://linkinghub.elsevier.com/retrieve/pii/0024379595001992 (visited on 10/11/2022).

Hardoon, David R, Janaina Mourao-Miranda, et al. (2007). “Unsupervised analysis of fMRI data using kernel canonical correlation”. In: *NeuroImage* 37.4, pp. 1250–1259.

Hardoon, David R, Sandor Szedmak, and John Shawe-Taylor (2004). “Canonical correlation analysis: An overview with application to learning methods”. In: *Neural computation* 16.12, pp. 2639–2664. DOI: 10.1162/0899766042321814.

Harris, Charles R et al. (2020). “Array programming with NumPy”. In: *Nature* 585.7825, pp. 357–362.

Hastie, Trevor et al. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer.

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.

— (1935). “Canonical correlation analysis (cca)”. In: *Journal of Educational Psychology*, p. 10.

— (1992). “Relations between two sets of variates”. In: *Breakthroughs in statistics*. Springer, pp. 162–190. DOI: 10.2307/2333955.

ICML (2023). *ICML 2023*. URL: https://icml.cc/Conferences/2023/Test-of-Time (visited on 09/21/2023).

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.

Jack Jr, Clifford R et al. (2008). “The Alzheimer’s disease neuroimaging initiative (ADNI): MRI methods”. In: *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine* 27.4, pp. 685–691.

James Chapman Janaina Mourao-Miranda, John Shawe-Taylor (2023). *A Framework for Regularised Canonical Correlation Analysis by Alternating Least Squares*.

Johnstone, Iain M (2001). “On the distribution of the largest eigenvalue in principal components analysis”. In: *The Annals of statistics* 29.2, pp. 295–327.

Kanai, Ryota and Geraint Rees (2011). “The structural basis of inter-individual differences in human behaviour and cognition”. In: *Nature Reviews Neuroscience* 12.4, pp. 231–242.

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.

Kingma, Diederik P and Jimmy Ba (2014). “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980*.

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.

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.

Lê Cao, Kim-Anh et al. (2008). “A sparse PLS for variable selection when integrating omics data”. In: *Statistical applications in genetics and molecular biology* 7.1.

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*.

Liu, Jingyu and Vince D Calhoun (2014). “A review of multivariate analyses in imaging genetics”. In: *Frontiers in neuroinformatics* 8, p. 29.

Liu, Zhangdaihong et al. (2022). “Improved Interpretability of Brain-Behavior CCA With Domain-Driven Dimension Reduction”. In: *Frontiers in Neuroscience* 16, p. 851827.

Lorenzi, Marco, Andre Altmann, et al. (2018). “Susceptibility of brain atrophy to TRIB3 in Alzheimer’s disease, evidence from functional prioritization in imaging

genetics". In: *Proceedings of the National Academy of Sciences of the United States of America* 115.12, pp. 3162–3167. DOI: 10.1073/pnas.1706100115. URL: https://hal.science/hal-01756811.

Lorenzi, Marco, Boris Gutman, et al. (2017). "Secure multivariate large-scale multi-centric analysis through on-line learning: an imaging genetics case study". In: *12th International Symposium on Medical Information Processing and Analysis*. Vol. 10160. SPIE, pp. 347–353.

Luo, Chunjie et al. (2018). "Cosine normalization: Using cosine similarity instead of dot product in neural networks". In: *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I 27*. Springer, pp. 382–391.

Lyu, Qi et al. (2021). "Understanding latent correlation-based multiview learning and self-supervision: An identifiability perspective". In: *arXiv preprint arXiv:2106.07115*.

Ma, Zhuang, Yichao Lu, and Dean Foster (2015). "Finding linear structure in large datasets with scalable canonical correlation analysis". In: *International conference on machine learning*. PMLR, pp. 169–178.

Mackay, David John Cameron (1998). "Introduction to monte carlo methods". In: *Learning in graphical models*. Springer, pp. 175–204.

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.

Matkovic, Andraz et al. (2023). "The contribution of diverse and stable functional connectivity edges to brain-behavior associations". In: *bioRxiv*, pp. 2023–11.

Matkovič, Andraž et al. (2023). "Static and dynamic fMRI-derived functional connectomes represent largely similar information". In: *Network Neuroscience* 7.4, pp. 1266–1301.

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, Rick A Adams, et al. (2022a). “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, James Chapman, Rick A. Adams, et al. (Aug. 2022b). “Canonical Correlation Analysis and Partial Least Squares for identifying brain-behaviour associations: a tutorial and a comparative study”. en. In: *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*. ISSN: 2451-9022. DOI: 10.1016/j.bpsc.2022.07.012. URL: https://www.sciencedirect.com/science/article/pii/S2451902222001859 (visited on 08/29/2022).

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.

Mills-Curran, William C (1988). “Calculation of eigenvector derivatives for structures with repeated eigenvalues”. In: *AIAA journal* 26.7, pp. 867–871.

Miranda, Lucas et al. (2021). “Systematic review of functional MRI applications for psychiatric disease subtyping”. In: *Frontiers in Psychiatry* 12, p. 665536.

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.

Oja, Erkki (1982). “Simplified neuron model as a principal component analyzer”. In: *Journal of mathematical biology* 15.3, pp. 267–273.

OpenAI (2021). *ChatGPT: A Large-Scale Generative Model for Open-Domain Chat*. https://github.com/openai/gpt-3.

Park, S, Eva Ceulemans, and Katrijn Van Deun (2023). “A critical assessment of sparse PCA (research): why (one should acknowledge that) weights are not loadings”. In: *Behavior Research Methods*, pp. 1–20.

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.

Paszke, Adam et al. (2019). “Pytorch: An imperative style, high-performance deep learning library”. In: *Advances in neural information processing systems* 32, pp. 8026–8037.

Pearl, Judea (2009). *Causality*. Cambridge university press.

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

Perekrestenko, Dmytro et al. (2018). “The universal approximation power of finite-width deep ReLU networks”. In: *arXiv preprint arXiv:1806.01528*.

Phan, Du, Neeraj Pradhan, and Martin Jankowiak (2019). “Composable effects for flexible and accelerated probabilistic programming in NumPyro”. In: *arXiv preprint arXiv:1912.11554*.

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.

Riffenburgh, Robert Harry (1957). “Linear discriminant analysis”. PhD thesis. Virginia Polytechnic Institute.

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.

Rypma, Bart and Mark D'Esposito (2001). "Age-related changes in brain–behaviour relationships: Evidence from event-related functional MRI studies". In: *European Journal of Cognitive Psychology* 13.1-2, pp. 235–256.

Sanger, Terence D (1989). "Optimal unsupervised learning in a single-layer linear feedforward neural network". In: *Neural networks* 2.6, pp. 459–473.

Sansone, Emanuele and Robin Manhaeve (2022). "GEDI: GEnerative and DIscriminative Training for Self-Supervised Learning". In: *arXiv preprint arXiv:2212.13425*.

Schmidt, Erhard (1907). "Zur Theorie der linearen und nichtlinearen Integralgleichungen: I. Teil: Entwicklung willkürlicher Funktionen nach Systemen vorgeschriebener". In: *Mathematische Annalen* 63.4, pp. 433–476.

Smith, Samuel L et al. (2021). "On the origin of implicit regularization in stochastic gradient descent". In: *arXiv preprint arXiv:2101.12176*.

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.

Snoek, Cees GM et al. (2005). "Mediamill: Exploring news video archives based on learned semantics". In: *Proceedings of the 13th annual ACM international conference on Multimedia*, pp. 225–226.

Somandepalli, Krishna et al. (2019). "Multimodal representation learning using deep multiset canonical correlation". In: *arXiv preprint arXiv:1904.01775*.

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.

Tenenhaus, Arthur and Michel Tenenhaus (2011). “Regularized generalized canonical correlation analysis”. In: *Psychometrika* 76.2, p. 257. DOI: 10.1007/s11336-011-9206-8.

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*.

Townsend, Florence, James Chapman, and James Cole (Nov. 2023). *florencejt/fusilli: Fusilli v1.0.0*. Version v1.0.0. DOI: 10.5281/zenodo.10228564. URL: https://doi.org/10.5281/zenodo.10228564.

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

Tuzhilina, Elena, Leonardo Tozzi, and Trevor Hastie (2023). “Canonical correlation analysis in high dimensions with structured regularization”. In: *Statistical modelling* 23.3, pp. 203–227.

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

Van Essen, David C et al. (2013). “The WU-Minn human connectome project: an overview”. In: *Neuroimage* 80, pp. 62–79.

Vapnik, Vladimir (1999). *The nature of statistical learning theory*. Springer science & business media.

Vaswani, Ashish et al. (2017). “Attention is all you need”. In: *Advances in neural information processing systems* 30.

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

Virtanen, Pauli et al. (2020). “SciPy 1.0: fundamental algorithms for scientific computing in Python”. In: *Nature methods* 17.3, pp. 261–272.

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. (2018). “Finding the needle in high-dimensional haystack: A tutorial on canonical correlation analysis”. In: *arXiv preprint arXiv:1812.02598*.

— (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.

Winkler, Anderson M et al. (2020). “Permutation inference for Canonical Correlation Analysis”. In: *arXiv preprint arXiv:2002.10046*.

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.

Wold, Herman (1975). “Path models with latent variables: The NIPALS approach”. In: *Quantitative sociology*. Elsevier, pp. 307–357.

Wong, YK (1935). “An application of orthogonalization process to the theory of least squares”. In: *The Annals of Mathematical Statistics* 6.2, pp. 53–75.

Yao, Yuan, Lorenzo Rosasco, and Andrea Caponnetto (2007). “On early stopping in gradient descent learning”. In: *Constructive Approximation* 26, pp. 289–315.

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*.

Zhang, Chiyuan et al. (2021). “Understanding deep learning (still) requires rethinking generalization”. In: *Communications of the ACM* 64.3, pp. 107–115.

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*.

Zou, Hui, Trevor Hastie, and Robert Tibshirani (2006). “Sparse principal component analysis”. In: *Journal of computational and graphical statistics* 15.2, pp. 265–286.

Zou, Hui and Lingzhou Xue (2018). “A selective overview of sparse principal component analysis”. In: *Proceedings of the IEEE* 106.8, pp. 1311–1320.