Learning Representations from Noisy Data and Brain Imaging: Subspace Modeling for Heteroscedastic Data and Deep Learning for Functional MRI in Alzheimer's Disease

by

Javier Salazar Cavazos

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical and Computer Engineering)
in the University of Michigan
2026

Doctoral Committee:

Associate Professor Laura Balzano, Co-Chair
Professor Jeffrey A. Fessler, Co-Chair
Professor Douglas C. Noll
Research Scientist Scott Peltier

Javier Salazar Cavazos
javiersc@umich.edu
ORCID iD: 0009-0009-1218-9836

ACKNOWLEDGMENTS

The PhD journey is a long one, filled with many highs and possibly even more lows, and my journey has been made much better by all the people I interacted with along the way. I want to take this opportunity to acknowledge the people who made this possible; since, "it takes a village" to get through the program, as Naveen would say.

Firstly, I thank my co-advisors, Professors Laura Balzano and Jeff Fessler, whom I met during visit week. I feel that the most important factor in a PhD student's ability to complete the program and excel is the advisor-student relationship. A big part of the reason why I joined the University of Michigan is that I got a good sense from both of you. You showed patience, gave me guidance to grow, and believed in me during the hardest of times. I was given freedom and space to explore interesting problems, which led me to become a better independent researcher, which is probably one of the main goals of the doctoral program. Doing a PhD in itself is already very challenging, and I cannot imagine what the experience would have been like with awful advisors. Thankfully, I never had to worry about this and just focused on doing good work.

Secondly, I thank my other co-advisor, Scott Peltier, for involving me in the wonderful world of functional MRI. My other work on heteroscedasticity in ML is more theoretical, so it was very fascinating to work in an applied field like Alzheimer's disease, which has very real implications. As a result, I got a very good mix of research work involving "model-to-data" and "data-to-model" problems during my time at UM. On another note, this subject is very meaningful to me personally due to having one family member who is experiencing mild cognitive impairment. We can only hope that our findings will lead to better intervention and mitigation strategies for this disease and cognitive decline.

Third, I would like to thank my undergraduate research advisor, Professor Gaik Ambartsumian from UT Arlington. I am the first in my family to attend college. I had no plans for graduate school, and I did not even know what research entailed. It was not until I did my first undergraduate research experience involving CT reconstruction that I realized how fun, creative, challenging, and rewarding research can be; all thanks to you. Your guidance and insights were instrumental in my early development as a person and as an academic.

Fourth, I am thankful for all my colleagues who made coming to campus more fun. I will miss our fun interactions. I thank the Lab of Jeff members {Jason, Amaya, Sofie, Robert, Rodrigo, Cyrus, Andi, Xiaojian, Caroline, Cameron, Naveen, Anish, Zongyu} for the camaraderie in our shared experience. Additionally, I thank SPADA members {Can, Rachel, Kyle, Alec, and Soo

Min} for the same reason. I want to highlight Naveen and Anish in particular from these lists. In many ways, both of you were my mentors early on in my journey. My first year was especially brutal, dealing with impostor syndrome, and you always knew the right things to say to help me keep going, push forward, and excel.

Fifth, I am deeply grateful to my family for their unwavering support, encouragement, and love throughout my journey. To my mom and dad, thank you for being devoted parents and always inspiring me to pursue my studies. To my wife, Beatrice, I owe immense thanks for your complete support and unconditional love, which enabled me to complete this program. And to my in-laws, you have been parents to me away from home, generously helping us navigate life's many hurdles and challenges.

Lastly, my graduate studies were supported by the Rackham Merit Fellowship, the National Science Foundation (NSF) CAREER Grant CCF-1845076, the NSF Grant CCF-2331590, the National Institutes of Health (NIH) grant R21 AG082204, and the National Institute on Aging (NIA) Grant P30 AG072931. Many thanks to these sources that gave me great academic freedom to explore problems and made this work possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Modern machine learning in signal processing, computer vision, and scientific domains increasingly seeks interpretable representations from large, unstructured data. This dissertation focuses on representation learning: discovering features of data that support downstream tasks such as classification and next-token prediction. Raw data are not always semantically meaningful on their own. For example, the word "apple" is ambiguous without context (fruit or company?), so models are needed to learn representations that better capture the underlying structure. Across its chapters, this thesis develops methods to learn such representations to enable improved analysis and inference.

Chapter 3 addresses a common issue in modern, messy datasets: heteroscedastic noise (noise with sample-dependent variance). Many traditional subspace learning methods degrade in this setting. We analyze how heteroscedasticity affects subspace estimation and propose ALPCAH, which downweights noisier samples to recover a more accurate low-dimensional subspace without knowledge of data quality. Experiments show ALPCAH outperforms standard approaches such as PCA, improving low-dimensional structure discovery for scientific data analysis.

Chapter 4 extends this idea from Chapter 3 to a union-of-subspaces setting, where each sample is assumed to lie near one of several, unknown low-dimensional subspaces. We introduce ALPCAHUS, an ensemble clustering approach that simultaneously clusters samples under heteroscedastic noise and estimates the corresponding multiple subspace bases. Results demonstrate improved clustering quality compared with existing methods, enabling more reliable group and structure discovery in applications involving heterogeneous, noisy data.

Chapter 5 focuses on deep clustering with imbalanced clusters. We extend TURTLE, a deep clustering method based on unsupervised support vector machines, by replacing its implicit uniform cluster-size assumption with a power-law prior on cluster proportions. The resulting method, PET-TURTLE, improves clustering performance in imbalanced regimes common in applications such as medical imaging, where positive cases may be rare.

The next chapters apply representation learning to Alzheimer's disease (AD), a growing public health challenge. Brain atrophy and functional changes can precede clinical symptoms, making early detection of affected neuro-circuitry crucial for predicting progression and informing treatment development. Chapter 6 studies AD diagnosis using resting-state fMRI and proposes a novel 4D convolutional neural network that operates directly on raw spatiotemporal data. This approach improves diagnostic performance compared to traditional 3D deep learning models.

The Montreal Cognitive Assessment (MoCA) is a clinical test of memory, language, and visuospatial skills used to detect mild cognitive impairment and dementia. Chapter 7 introduces NeuroMamba, a deep state space model designed to uncover brain–behavior relationships in resting-state fMRI by predicting MoCA scores. NeuroMamba achieves stronger associations with MoCA than connectivity-based baselines and highlights brain regions implicated in AD, which may inform targeted brain stimulation studies. Additionally, Sec. 7.7 presents preliminary results on MoCA prediction from task-based fMRI, including face–name and object–location association tasks.

CHAPTER 1

Introduction

1.1 Big Data

Big data has increasingly become essential to modern business, science, and technology. The widespread use of smartphones, cloud servers, and consumer-grade hardware now generates vast amounts of data that are inexpensive to collect and easy to store at scale. Alongside this, recent advancements in machine learning have transformed how we use massive datasets to identify key patterns and predict outcomes in various industries and scientific fields. Large, data-driven models fuel business recommendation engines and scientific computing applications, such as medical and astronomical imaging, that can identify important signals from unstructured data. As the information age continues, the frontiers of modern data science involve measuring data across multiple modalities, integrating data from diverse sources, and handling a range of processing tasks, such as subgroup discovery. Handling modern data presents unique challenges, especially when starting with large volumes of raw data. The initial step of any machine learning pipeline typically involves preprocessing "messy" data, as entries may be noisy, severely flawed, or even missing. Combining data from a wide range of sources increases the number of samples available for constructing large models, but it also introduces the challenge of managing heterogeneous data. Or, in another instance, all data comes from one machine but may vary in quality due to physical phenomena, such as atmospheric perturbations, or may be purposely altered to limit radiation in a medical imaging context. Regardless, in many typical machine learning models, such heterogeneity can significantly impair model performance.

In certain situations, we need to preprocess messy data to make it suitable for subsequent machine learning and data science tasks. For instance, in astronomy quasar applications, it is essential to denoise data to more accurately estimate the principal components of uncorrelated wavelength and frequency patterns of different quasar groups. Alternatively, the goal may be to infer missing entries, detect outliers, or perform some other task. Take the "Netflix challenge" [5] for example, where recommender systems are used to learn user preferences for movie and TV

show recommendations. Here, user ratings create a significantly incomplete matrix across millions of users and thousands of products, such as movies and TV shows. Since each user rates only a small subset of titles, most of the data is missing. A low-rank model of the data can predict these missing ratings to suggest new titles [6]. These problems can be solved using subspace learning algorithms that find a compact, lower-dimensional representation of the data.

In many situations like those above, one or more sensors gather high-dimensional data, meaning data with a large number of measured variables across numerous samples. Even though the total number of measurements is substantial, we often aim for a simplified representation of the data that compresses large, cumbersome datasets into a concise model, identifying patterns or structures that facilitate scientific investigation and use this underlying structure for classification, prediction, and other tasks. Such low-dimensional representations enable us to carry out essential signal processing tasks as described earlier. Since much of the data exhibits patterns with a limited number of degrees of freedom that change slowly or in a restricted manner, it can often be effectively modeled using low-dimensional subspaces. The first few chapters of this thesis focuses on low-rank structures for this noisy, heteroscedastic data.

1.2 Functional Magnetic Resonance Imaging

Functional magnetic resonance imaging (fMRI), which captures a series of MRI scans of the brain, has been increasingly used to gain insights into brain function. It measures neural activity by indirectly using the blood-oxygen-level-dependent (BOLD) signal. This neural activity affects local blood flow in the brain, which in turn influences the MRI signal. There are two primary types of fMRI acquisition: task-based and resting-state. Task-based fMRI requires the subject to perform a specific task during the scan, whereas resting-state fMRI is conducted without any external task stimulus. Task-based fMRI is used for specific purposes, for example, finger tapping, which is commonly used as a test case to see activation of the motor cortex region.

Resting-state fMRI measures intrinsic brain activity when the subject is not engaged in any specific task. Studies have shown that the default-mode network is active in the brain's baseline state, such as during wakeful rest, self-referential processing, and internally directed thinking [7] [8]. Resting-state scans are particularly useful in studies involving groups that may struggle with task completion due to conditions affecting motor control, attention, or cognition. Functional MRI is highly dimensional since a 3D volume is acquired over time. One can use this raw spatial-temporal data, but due to computational constraints, it is also beneficial to explore alternative approaches. As a result, there are many ways to extract relevant signal data from fMRI.

Firstly, timeseries data can be extracted to reduce the raw (X, Y, Z, T) space to (B, T), where B time signals of length T are obtained from B brain regions. There are several methods for

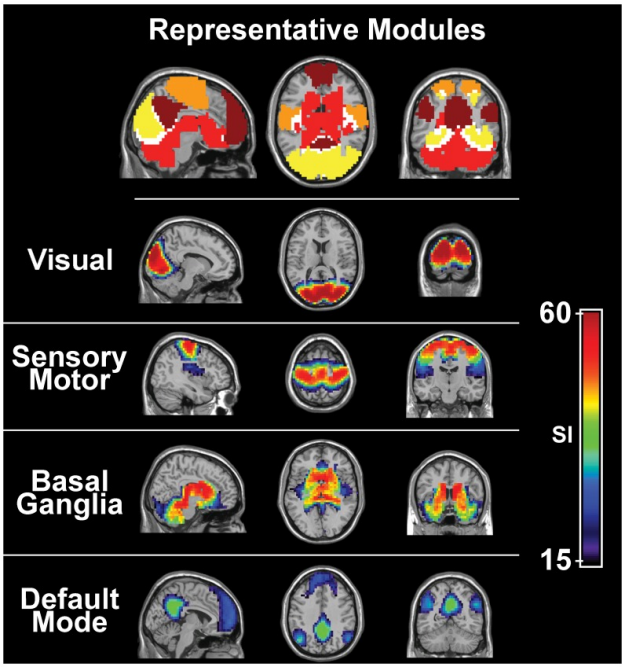


Figure 1.1: Study showing four functional networks that were found to be highly consistent across subjects. These modules include the visual (yellow), sensory/motor (orange) and basal ganglia (red) cortices as well as the default mode network (precuneus/posterior cingulate, inferior parietal lobes, and medial frontal gyrus; maroon). Figure taken from [1].

identifying brain regions to evaluate functional activity. One can use a region of interest derived from a predefined brain atlas, segmentation template, or a brain parcellation algorithm to find natural clusters of similar BOLD activity, e.g., applying K-means on all the timeseries to group regions. In these approaches, preprocessed fMRI time courses are averaged across each ROI or parcel. Additionally, there are decomposition methods, such as principal component analysis, that are less interpretable but can find suitable subspaces for dimensionality reduction.

Secondly, functional connectivity matrices can be extracted to reduce the (X, Y, Z, T) space to (B, B) involving B regions. Functional connectivity refers to the spatial relationship between fMRI BOLD signals in different brain regions. This relationship is often measured using correlation, and regions that exhibit coordinated variation are believed to have direct or indirect communication. It has been demonstrated that functional connectivity in rs-fMRI shows regions of interest in the motor cortex that were significantly more correlated with each other than with other parts of the brain in humans [9]. This finding was quickly replicated across other domains, such as language [10] [11]. Importantly, connectivity can be examined using both task-based and resting-state fMRI, each with unique advantages. Functional connectivity can be calculated as a static measure by using the entire time series for correlation or as a dynamic measure by dividing the time series into multiple states and deriving connectivity metrics for each state. This approach

allows researchers to explore how connectivity changes over time across the brain. Over the years, there has been a growing interest in MRI-based functional research. These methods are attractive for neurofunctional studies because they are noninvasive and offer indirect measurements of brain activity that capture both whole-brain and temporal information.

The later chapters of this thesis concentrates on using functional MRI data to develop computational models for Alzheimer's disease diagnosis and behavior score prediction. In Sec. 7.8.1, we merge heteroscedastic data and functional MRI topics together to show potential applications of heteroscedastic subspace learning for motion estimation and correction in functional MRI, linking the main ideas of this thesis under the branch of representation learning.

1.3 Contributions & Organization

The rest of this thesis is organized as follows:

- Chapter 2 contains relevant background information on subspace learning, subspace clustering, functional MRI, Alzheimer's disease, and behavior scores.
- Chapter 3 **(ALPCAH: Subspace Learning for Sample-wise Heteroscedastic Data)** develops a subspace learning algorithm for heteroscedastic sample-wise data. This work appeared in conference proceedings in Sampling Theory and Applications 2023 [12] and published in IEEE Transactions on Signal Processing (TSP) [13].
- Chapter 4 **(ALPCAHUS: Subspace Clustering for Heteroscedastic Data)** extends Chapter 3 by developing a subspace clustering algorithm, in the union of subspaces setting, for heteroscedastic sample-wise data. It is under review at IEEE Transactions on Signal Processing (TSP) [14].
- Chapter 5 **(PET-TURTLE: Deep Unsupervised Support Vector Machines for Imbalanced Data Clusters)** introduces a deep clustering algorithm, inspired by the unsupervised support vector machine algorithm named TURTLE, that better handles imbalanced data distributions. This has been published in IEEE Signal Processing Letters [15].
- Chapter 6 **(Alzheimer's Disease Classification in Functional MRI via 4D Convolutions)** develops a novel 4D convolutional neural network that learns temporal-spatial kernels for Alzheimer's disease diagnosis in resting-state functional MRI. This work has appeared in the medical conference called International Society for Magnetic Resonance in Medicine (ISMRM) 2025 [16].

- Chapter 7 (**Behavior Score Prediction in Resting-State and Task-Based Functional MRI**) explores behavior score prediction in resting-state functional MRI by deep state modeling, inspired by the Mamba architecture, to learn from temporal dynamics as opposed to working with functional connectivity. This work will be submitted to IEEE Journal of Biomedical and Health Informatics [17]. Some preliminary results on task-based functional MRI are included in this chapter. Further work is necessary for the publication of these results associated with face-name and object-location association tasks.
- Chapter 8 contains ideas for future work related to these chapters.

CHAPTER 2

Background

This section discusses some background information relating to the thesis, namely, low-rank modeling, such as subspace learning and clustering, functional MRI, and Alzheimer's disease.

2.1 Low-Dimensional Models

This subsection discusses some background information relating to heteroscedastic data, subspace learning for low-dimensional modeling, and union of subspaces modeling for clustering.

2.1.1 Heteroscedasticity

What if our data comes from medical imaging devices with varying radiation levels? What if we capture astronomical features at different time points with fluctuating atmospheric conditions? Or what if we combine high-precision government-grade air quality measurement sensors with data from low-cost consumer-grade sensors?

Modern datasets are increasingly large and formed by merging heterogeneous samples from diverse sources or conditions, often exhibiting heteroscedastic noise, meaning noise with varying variances. For instance, environmental monitoring often involves combining data from a few high-precision instruments with a large volume of crowd-sourced data from consumer hardware. Data heterogeneity is prevalent across many imaging applications, ranging from medical instruments to astronomical research. In machine learning applications involving computed tomography (CT) images, high-dose images have better signal-to-noise ratios and clearer reconstructions than low-dose images. Additionally, magnetic resonance imaging (MRI) machines with multiple coils produce images with nonuniform noise from each coil. In astronomy, images collected across different nights can vary in quality due to atmospheric turbulence and the distortion of light from distant stars and galaxies.

The first part of this thesis aims to address the problem of data with varying quality, specifically, additive heteroscedastic noise in which noise levels differ across samples. Current algorithms, such as Principal Component Analysis (PCA) [18], assume that all data share the same

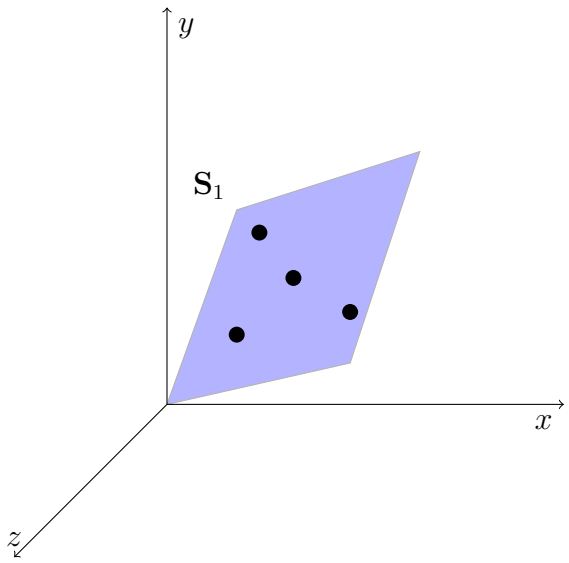


Figure 2.1: A 2D subspace in a 3D ambient space.

noise variance, which makes them sensitive when encountering outliers or samples with higher noise variances. Addressing data heterogeneity is an important challenge in machine learning, requiring advanced models that account for it.

2.1.2 Subspace Learning

Principal component analysis (PCA) [18] is a linear technique for reducing dimensionality, commonly used in exploratory data analysis, visualization, and data preprocessing. It involves linearly transforming the data into a new coordinate system so that the directions (principal components) representing the most significant variation are easily identifiable. For a set of points in real coordinate space, the principal components are a sequence of unit vectors, where each i-th vector indicates the direction of a line that best fits a modified version of the data, where we have removed components $(1, \ldots, i-1)$, while being orthogonal to the first $i-1$ vectors. The best-fitting line is one that minimizes the average squared distance from the points to the line. These directions form an orthonormal basis in which each dimension of the data is linearly uncorrelated with the others. PCA is applied in numerous fields; it is impossible to list all of them, but some applications include population genetics, microbiome research, and astronomy. Genuinely, it would be surprising if there exists a computational field that does not use PCA in some capacity. See Fig. 2.1 for a visualization example of a 2D subspace in a 3D ambient space. Bear in mind that most data is extremely big, beyond 3 dimensions, as a simple $256 \times 256 \times 3$ RGB image contains about $200,000$ dimensions in this case.

Consider a dataset $\boldsymbol{Y}$ with N samples and D features such that $\boldsymbol{Y} \in \mathbb{R}^{D \times N}$ and the columns of $\boldsymbol{Y}$ contain $\boldsymbol{y}_i \in \mathbb{R}^D$, D-dimensional data vectors. The first step involves centering the data by subtracting the mean of each feature such that

$$\boldsymbol{Y} = \boldsymbol{Y} - \mathbb{1}\boldsymbol{\mu}' \tag{2.1}$$

where $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{y}_i$ is the sample mean. This is done since a subspace always contains the origin and would lead to suboptimal results if the data is not centered. Conceptually, one may find the first principal component that maximizes the variance by solving

$$\hat{\boldsymbol{u}}_1 = \arg\max_{\boldsymbol{u}} \frac{1}{N}\sum_{i=1}^{N} |\langle \boldsymbol{y}_i, \boldsymbol{u}\rangle|^2 \text{ s.t. } \|\boldsymbol{u}\|_2 = 1. \tag{2.2}$$

That is, finding the vector whose dot product is maximal with all data samples while remaining of unit norm. In general, one may find all principal component vectors in matrix form $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_D]$ by solving

$$\hat{\boldsymbol{U}} = \arg\max_{\boldsymbol{U}} \text{Trace}(\boldsymbol{U}^T\boldsymbol{Y}\boldsymbol{U}) \text{ s.t. } \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}. \tag{2.3}$$

Trace maximization problems with orthogonal constraints are well-studied in the literature and are known to involve solutions involving Singular Value Decomposition (SVD). In this case, an SVD on the data matrix $\boldsymbol{Y}$ will return the left and right singular vectors and singular values associated with the matrix. Mathematically, $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}' = \sum_{i=1}^{D} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i'$. One must select the number of left singular vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_d$ associated with singular values $\sigma_1, \ldots, \sigma_d$ where $d \ll D$ to find a d-dimensional subspace $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_d] \in \mathbb{R}^{D \times d}$. This can be done in many ways, a scree plot being the most common way to select d. The idea being that if $d = 3$ captures 95% of the variance of the 300-dimensional data, then it may be a suitable value to use since most of the "essence" of the data is captured. The subspace coefficients or coordinates can be obtained by projecting the data matrix onto that subspace, i.e., $\boldsymbol{U}'\boldsymbol{Y}$, for further analysis, such as clustering, denoising, data compression, and other techniques.

2.1.3 Cluster Analysis

Cluster analysis, or clustering, involves organizing a collection of objects in such a manner that items within the same group (or cluster) share greater similarity, based on some criteria, than those in different groups. It serves as a key method in exploratory data analysis and is widely applied in statistical data analysis across various disciplines such as pattern recognition, image analysis, information retrieval, bioinformatics, data compression, computer graphics, and machine learning.

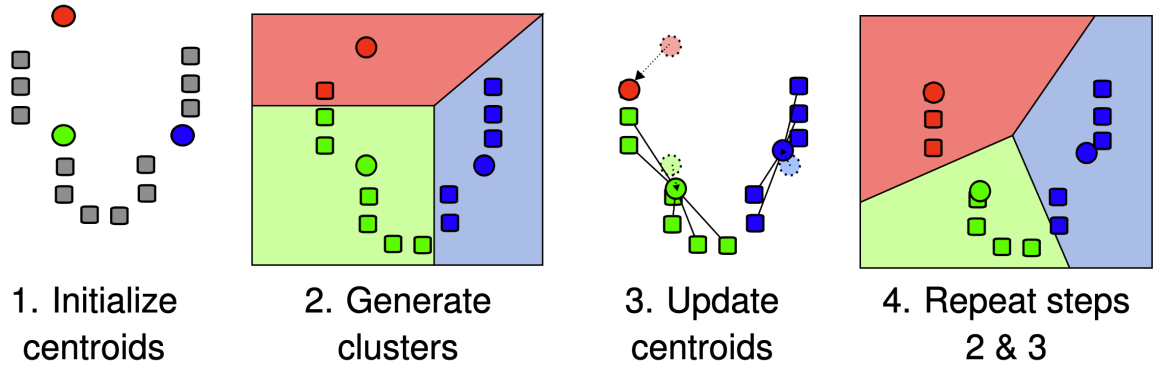


Figure 2.2: K-means algorithm procedure used to find 3 clusters associated with the gray square points (dataset) using computed centroids in circle shape. Colors represent current cluster assignment estimates in the iterative algorithm. Attribution: Weston.pace, K-Means Example Step 2, CC BY-SA 3.0.

For example, in bioinformatics, many biologists are interested in identifying gene groups that share common expressions across cells. Clustering can be used in this instance to identify gene groups that lead to interesting biological conclusions about the role of genes in explaining observable traits.

Rather than being a singular algorithm, cluster analysis encompasses a variety of algorithms and tasks. These algorithms vary greatly in their definitions of what constitutes a cluster and their methods of identifying them. Common interpretations of clusters include groups with minimal distances between members, dense regions in the data space, or segments fitting specific statistical distributions. As such, clustering can be approached as a multi-objective optimization problem. The choice of a suitable clustering algorithm and its parameters, such as the distance metric, density threshold, or expected number of clusters, is influenced by the specific dataset and the analysis goals. Consequently, cluster analysis is not a fully automated process but an iterative one that involves knowledge discovery and optimization. It often requires adjustments to data preprocessing and model parameters to achieve the desired outcomes.

The concept of a "cluster" lacks a precise definition, which contributes to the existence of numerous clustering algorithms. The common thread among these algorithms is the grouping of data objects. However, different researchers adopt varying cluster models, and each model can be associated with different algorithms. The characteristics of clusters identified by different algorithms can vary widely, making it important to understand these cluster models to grasp the distinctions between the algorithms. Typical cluster models include:

- Connectivity models: For instance, hierarchical clustering [19] creates models based on distance connectivity using any distance metric.

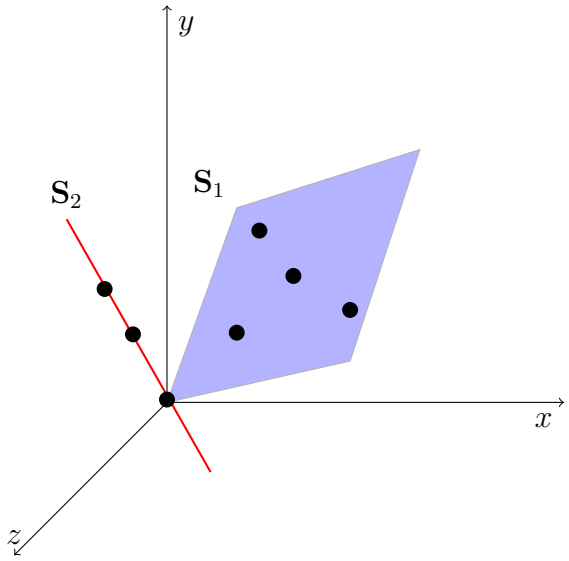


Figure 2.3: A 2D subspace and 1D subspace in a 3D ambient space.

- Centroid models: The K-means algorithm [20], arguably the most ubiquitous clustering method, represents each cluster with a single mean vector and alternates between updating the centroids and reassigning points to each cluster. See Fig. 2.2 for a visualization of this algorithm.
- Distribution models: In this approach, clusters are formed using statistical distributions, such as the multivariate normal distribution used by the expectation-maximization algorithm.
- Density models: Algorithms like DBSCAN [21] and OPTICS [22] define clusters as connected dense regions within the data space.
- Subspace models: In this modeling, the data is assumed to lie in a union of subspaces, with each cluster belonging to a distinct low-dimensional subspace. Some example methods are Sparse Subspace Clustering [23] and K-Subspaces [24].

This is only a subset of possible clustering algorithms. The work in this thesis primarily concerns subspace models, so the discussion will shift to subspace clustering.

2.1.4 Union of Subspaces (UoS) Clustering

Union of Subspaces (UoS) models generalize subspace learning when multiple subspaces exist, as depicted in Fig. 2.3. This more general setting allows one to model data that is more complex to be captured by multiple subspaces. A practical example could be that each subspace corresponds to a class. Let $\mathcal{U} = \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_K$ be a union of K subspaces where $\mathcal{S}_i \subset \mathbb{R}^D$ are d_i-dimensional

subspaces of the ambient space $\mathbb{R}^D$. The goal is to recover the UoS $\mathcal{U}$ from data samples $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N]$ while simultaneously clustering each data sample into one of the $\mathcal{S}_i$ subspaces. If the cluster labels were known, say classes of plant species, then $\mathcal{U}$ is easily learnable by applying PCA on each cluster independently. Likewise, if one knew $\mathcal{U}$, then the cluster labels are easily discovered by finding the nearest subspace for each data sample. However, in many situations, both are missing, making them harder to find. Hence, subspace clustering aims to simultaneously find both the cluster labels and subspaces.

2.2 Brain Activity & Alzheimer's Disease

This subsection discusses some background information relating to Alzheimer's disease, functional MRI, and behavior scores.

2.2.1 Alzheimer's Disease

Alzheimer's disease (AD) is a progressive neurodegenerative disorder that often leads to dementia, characterized by a decline in cognitive and memory functions. About 6 million Americans are currently affected by AD, and this figure is expected to rise to 12.7 million by 2050 due to the aging population [25]. Recently, three distinct stages of Alzheimer's disease have been recognized clinically: the preclinical stage, an intermediate stage known as mild cognitive impairment (MCI), and, in advanced stages, dementia of the Alzheimer's type (DAT) [26]. Patients who show no symptoms or signs of Alzheimer's disease are known to be cognitively normal (CN). MCI is diagnosed when there is objective evidence of cognitive disturbances, despite the relative preservation of daily functioning. DAT is characterized by severe cognitive and functional impairments that require clinical diagnosis. Patients with MCI, the earliest clinical phase of DAT, are at a significant risk of developing dementia, though the probability and speed of this progression vary among individuals. Additionally, preclinical AD is known to impact the brain years before any diagnosis. As a result, there is a growing need to study brain changes in the early stages to support future research on detection, prediction, and treatment strategies.

Dementia is a general term for cognitive and memory deficits. Several diseases or disorders can lead to dementia, with Alzheimer's disease being one of the most common causes. Other contributors include cerebrovascular disease, Lewy body disease, frontotemporal lobar degeneration, and Parkinson's disease, among others. However, coexisting conditions or mixed pathologies often occur between these causes, making diagnosis more complex [27]. This complexity underscores the need for adaptable biomarkers that can differentiate between or highlight overlapping pathologies when applied across different diagnoses.

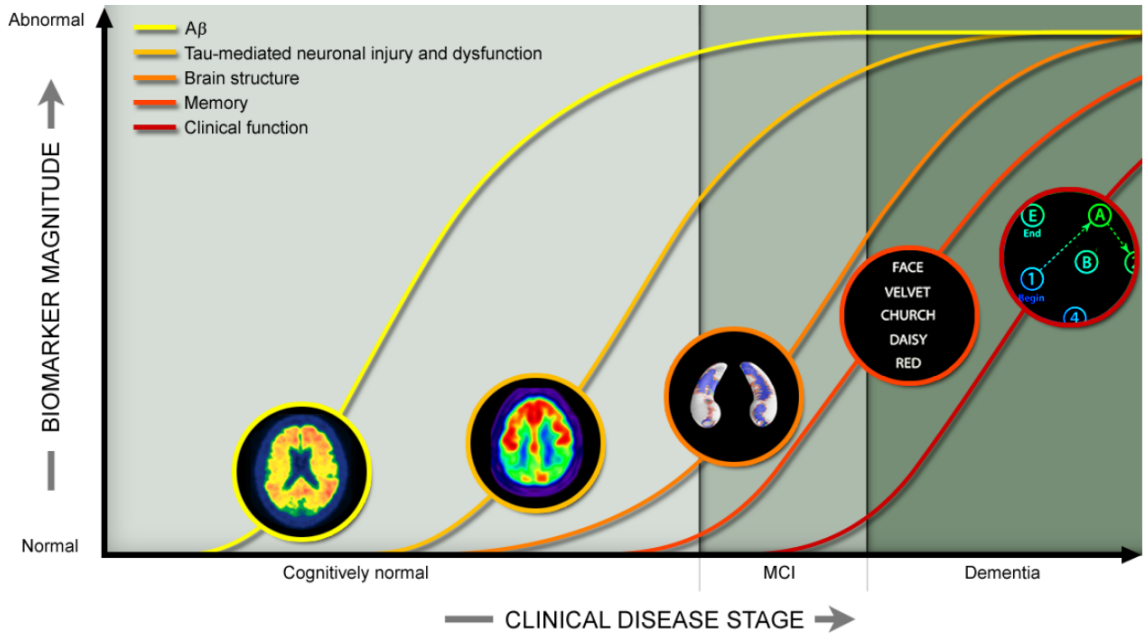


Figure 2.4: Schematic of Alzheimer's disease biomarkers and their progression over time. Figure from the Alzheimer's Disease Neuroimaging Initiative [2].

Alzheimer's disease pathology is characterized by the accumulation of amyloid-β and tau proteins, which begins years or even decades before a diagnosis is made, as indicated in Fig. 2.4. The presence of tau is associated with cognitive decline. The hippocampus, located in the temporal lobe, is one of the first brain structures affected by Alzheimer's disease. The Braak staging model describes the progression of protein accumulation, starting in the inferior temporal and medial frontal lobes and eventually affecting most areas of the brain [28]. This buildup is followed by nerve cell degeneration, or brain atrophy, which can be seen in brain imaging (such as MRI) as enlargement of the ventricles, widening of the sulci, and thinning of the gyri [29]. At this stage, cognitive and behavioral changes start to appear as the disease advances.

Cognitive changes are a natural part of healthy aging, but they become more pronounced with conditions such as MCI and DAT. Episodic memory, which involves the conscious recall of detailed long-term memories of unique past events, is a well-known area affected in dementia and is often associated with the default-mode network [30]. However, episodic memory performance is also expected to decrease with age. On the other hand, semantic memory, which encompasses general knowledge of the world, tends to remain stable throughout life and may help differentiate between aging and pathological cognitive decline [31]. Still, it may not be consistently impaired across individuals on the Alzheimer's disease spectrum [32]. Other cognitive changes that may arise with age or dementia include declines in spatial abilities, reasoning, and processing speed. There is growing interest in distinguishing between normal aging and pathology because Alzheimer's

disease causes brain changes many years before clinical symptoms appear, and identifying early functional brain changes from those of normal aging can aid early detection.

Biomarkers for Alzheimer's disease primarily target three key components: amyloid-β, tau pathology, and neuronal injury [33] [34]. To observe amyloid and tau in vivo, cerebrospinal fluid (CSF) measures and positron emission tomography (PET) are used. PET ligands, such as fluorodeoxyglucose (FDG), have become established biomarkers for Alzheimer's. FDG is used to evaluate glucose metabolism in the brain, which is often irregular in AD [35]. FDG-PET typically shows reduced metabolism in regions with brain atrophy [36] [37]. Additionally, FDG and other PET ligands are employed to study tau pathology in living patients. Neuronal injury is evaluated using structural MRI, evidence of hypometabolism in FDG-PET scans, or by measuring total tau in CSF. Structural MRI has detected AD-related brain atrophy up to ten years before the onset of symptoms and a formal diagnosis [38]. Alzheimer's disease is suspected when biomarker data and clinical cognitive evaluations point to AD pathology, which can be confirmed post-mortem through neuropathological studies using Pittsburgh Compound-B to identify amyloid-β deposits in the brain [39]. There is a clinical need for new biomarkers to assess various aspects of Alzheimer's disease and detect brain changes at earlier stages.

Increasing knowledge about the genetic aspects of Alzheimer's disease has emerged. The initial finding in this area was the link between the apolipoprotein E ϵ4 allele and an increased risk of developing Alzheimer's disease [40]. Further research has identified other genetic loci associated with the condition [41] [42], and there are notable correlations between genetic risk scores and the future risk of developing Alzheimer's, as well as the progression from MCI to DAT [43].

Increasing evidence supports the consensus that interventions should target the earliest stages of Alzheimer's disease. This underscores the need for advanced data acquisition and analysis methods to identify early brain changes, thereby aiding the discovery of biomarkers and enhancing disease detection and prediction. Research can use imaging techniques, cognitive assessments, and other tools to examine the structural and functional alterations associated with Alzheimer's disease. Functional MRI is particularly promising due to its noninvasive nature and its ability to integrate with other modalities, such as structural MRI and Diffusion Tensor Imaging (DTI).

2.2.2 Functional MRI in Alzheimer's Disease

Resting-state fMRI functional connectivity has been employed in numerous studies on Alzheimer's disease. The hippocampus, an area affected in the early stages of Alzheimer's, has been shown to exhibit disrupted resting-state functional connectivity in individuals with amnestic MCI [44]. Additionally, disrupted functional connectivity in the default mode network (DMN) is commonly

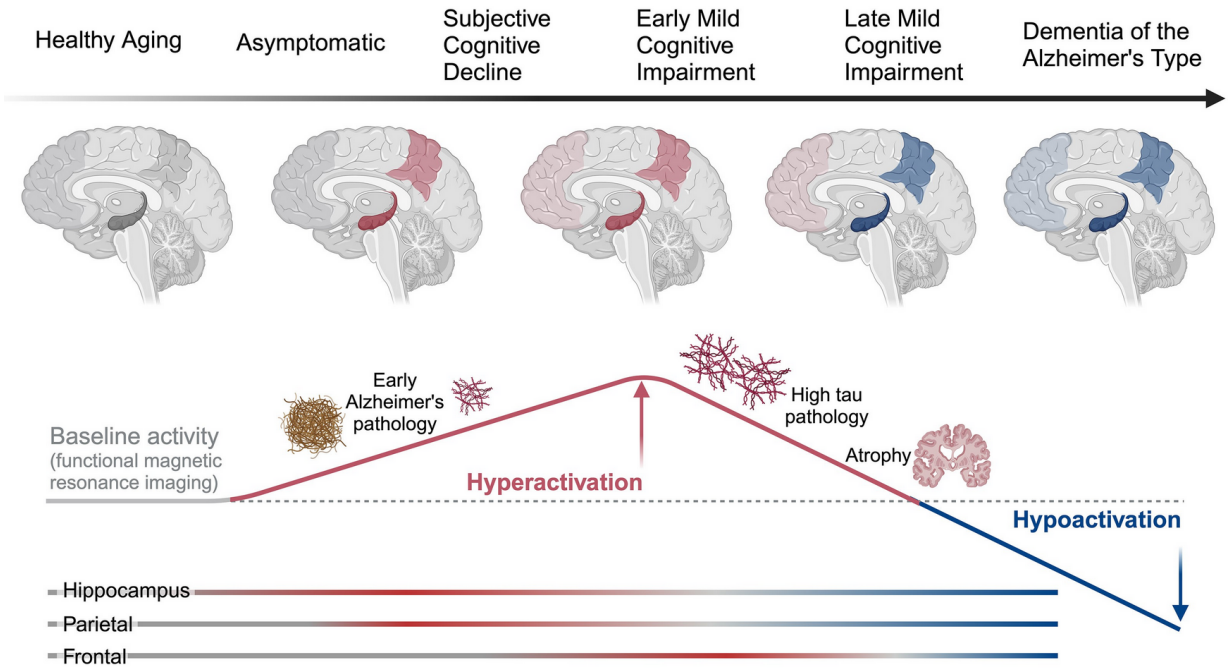


Figure 2.5: Brain activity for Alzheimer's disease subjects in functional MRI. Figure taken from [3].

observed in groups along the Alzheimer's disease spectrum. Changes in DMN connectivity have been noted in MCI and DAT patients compared to healthy controls [45] [46] [47] [48]. Impaired memory function is frequently associated with disrupted DMN functional connectivity [49] [50], and individuals with DAT exhibit reduced connectivity in the posterior DMN compared to healthy older adults [30].

Functional connectivity reveals significant differences between individuals with MCI and CN individuals, indicating that widespread degradation of brain networks can be detected in the early stages (MCI) [51]. In cases of amnestic MCI, researchers have noted initial increases in connectivity within the posterior cingulate cortex (PCC), which are followed by decreased PCC activity and heightened connectivity in the frontal network over time [52]. This pattern may indicate initial hyperactivation as a compensatory mechanism in the early stages of the disease, transitioning to hypoactivation as the pathology progresses. This progression often mirrors the pathological changes, beginning in the medial temporal lobe, spreading through regions of the DMN, including the PCC, and eventually reaching the frontal areas of the brain in later stages [53].

Understanding the neural mechanisms involved in early brain changes associated with AD is vital for predicting the progression to advanced stages and researching therapeutic interventions. The fact that functional connectivity is a sensitive indicator of memory and other Alzheimer related changes in the brain, coupled with evidence of increased functional connectivity in the initial

stages, suggests that it could be useful for early detection and may offer additional benefits for prediction studies.

Neuroimaging has become a crucial tool in the clinical assessment of individuals suspected of having neurodegenerative diseases like Alzheimer's. Structural MRI brain scans can reveal the presence and progression of neurodegeneration. Individuals with MCI and DAT often show the characteristic progressive atrophy associated with AD, particularly in regions such as the medial temporal lobes [29]. However, by the time significant brain atrophy is detected, the disease may have already been affecting the brain for years or even decades, making early detection essential. Current research focuses on better characterizing the early functional changes in the brain linked to AD, with functional MRI being one approach, as it serves as a proxy for neural activity.

Functional MRI measures are increasingly being studied as potential biomarkers for AD, with a focus on connectivity and network analysis. As noted, compensatory increases in fMRI activations during the early stages of Alzheimer's pathology are widely recognized, while general reductions in brain activity are often observed during advanced stages. Differences in fMRI activations might be associated with altered neural activity leading to impairments, such as memory deficits, or with neurovascular dysfunction affecting neurovascular coupling, among other possible reasons. The key takeaway is that fMRI can provide an indirect assessment of neuronal functioning and may help identify patients at risk of developing AD before significant atrophy occurs. Thus, detection and interventions are critical at the earliest stages of the disease, and fMRI is a promising tool due to its noninvasive nature and its ability to integrate with other modalities, potentially illuminating neural mechanisms underlying early changes and enabling improved early detection.

2.2.3 Task-based Functional MRI in Alzheimer's Disease

Task-based fMRI (tb-fMRI) differs from resting-state fMRI (rs-fMRI) because patients perform specific tasks while in the scanner. For example, a common task is face-name association, where a patient is shown a person's face along with their name. After a period of time (null period), the patient selects the correct name from a multiple-choice question. This test is meant to "stress test" a person's recall ability and the associated brain regions responsible for short-term recall. This contrasts with rs-fMRI, where a patient simply rests in the scanner. In relation to Alzheimer's disease, tb-fMRI is an underexplored area of research. A few small-sample studies have examined participants with dementia and used manual analyses to identify tb-fMRI differences compared with healthy individuals [3] [54]. From their findings, there appeared to be hyperactivation (longer time period of activity) in the early stages of MCI and hypoactivation (shorter time period of activity) in dementia subjects for the hippocampus, parietal, and frontal areas of the brain. Additionally,

late-stage MCI subjects saw both hyperactivation in the frontal region and hypoactivation in the hippocampus and parietal regions.

Surprisingly, there are no Alzheimer's disease datasets openly available for task-based fMRI. Because of ADNI (Alzheimer's Disease Network Initiative), both structural MRI and resting-state fMRI data are plentiful, leading to abundant research on AD using these data types. However, tb-fMRI is not included in any ADNI study, leading to a general lack of publications involving tb-fMRI and machine learning for AD applications. A recent collaboration with MADRC, the Michigan Alzheimer's Disease Research Center, resulted in a dataset in our collection that includes both resting-state and task-based fMRI data, enabling us to further explore the relationship between Alzheimer's disease and fMRI. In Chapter 7, we explore machine learning approaches using both rs-fMRI and tb-fMRI to uncover unusual patterns and spatial relationships in Alzheimer's subjects.

2.2.4 Behavior Scores

Behavior scores, in medical settings, refer to standardized tools used to assess cognitive, linguistic, and behavioral functioning, often guiding diagnoses and treatment in neurology, psychiatry, and geriatrics. In the context of fMRI research, these scores help quantify participants' abilities and link behavioral performance to neural activity. Examples include the Montreal Cognitive Assessment (MoCA) [4], which screens for mild cognitive impairment by evaluating memory, attention, and language with tasks like vegetable and animal naming; the Mini-Mental State Examination (MMSE) [4], a widely used brief tool measuring orientation, recall, and attention; and the MINT (Multilingual Naming Test) [55], specifically designed to assess language and naming ability across multiple languages. See Fig. 2.6 for an example of a subset of the questions asked during the MoCA exam. These and other assessments, such as verbal fluency tests and the Boston Naming Test, provide objective metrics for assessing cognitive impairment. These tests are easy and inexpensive to implement in practice, allowing clinicians to quickly obtain an initial assessment of cognitive impairment. The MoCA metric achieves an $\sim 89\%$ accuracy score when classifying healthy and mildly impaired individuals [4]. In the fMRI context, it can correlate with observed brain activation patterns, enabling researchers and clinicians to better understand neurocognitive deficits and which regions are most susceptible to impairment. In Chapter 7, we use MoCA along with composite z-score metrics for memory and language to augment our prediction problem.

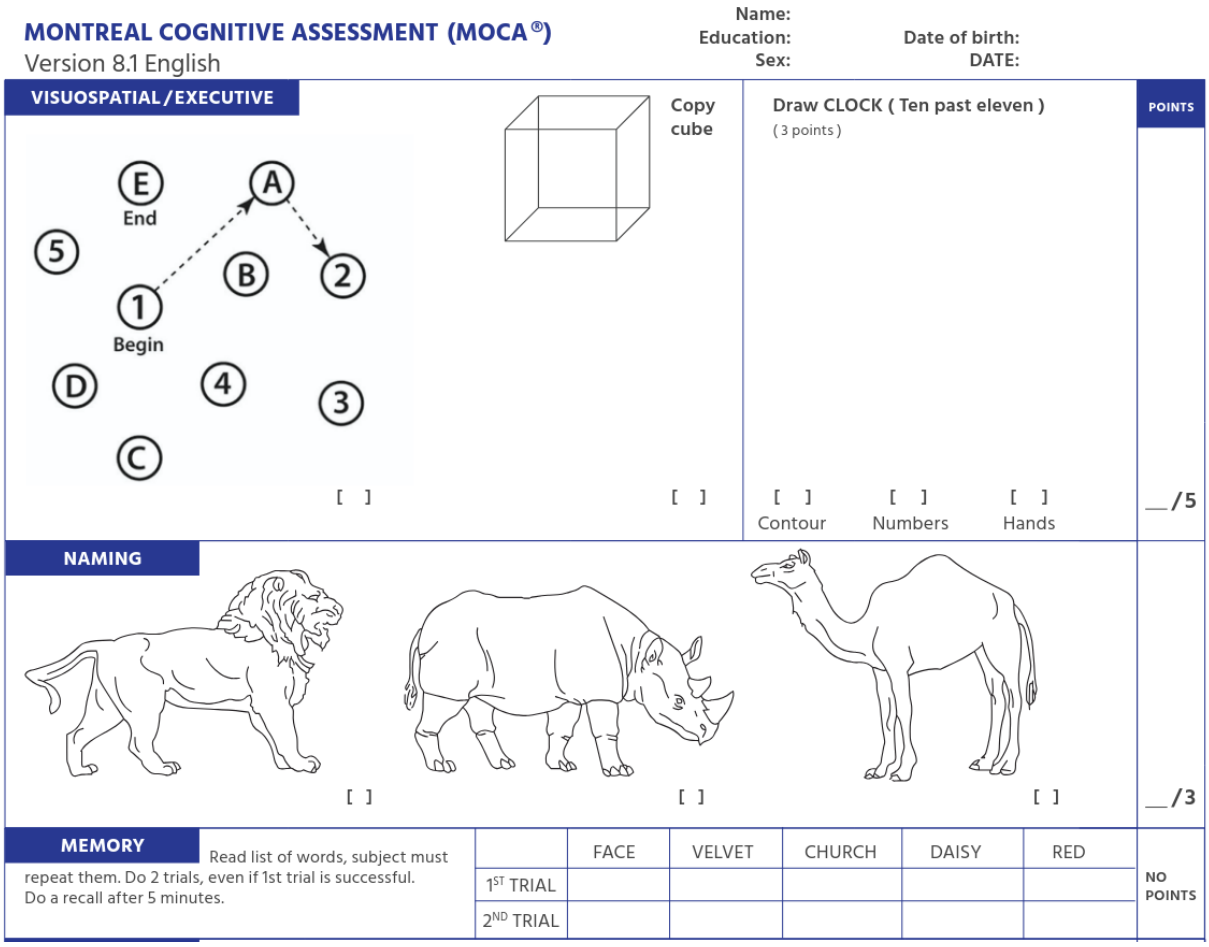


Figure 2.6: A subset of questions in the MoCA exam that pertain to visual-spatial skills, naming, and memory tasks used for cognitive impairment assessment. Figure taken from [4].

CHAPTER 3

ALPCAH: Subspace Learning for Heteroscedastic Data

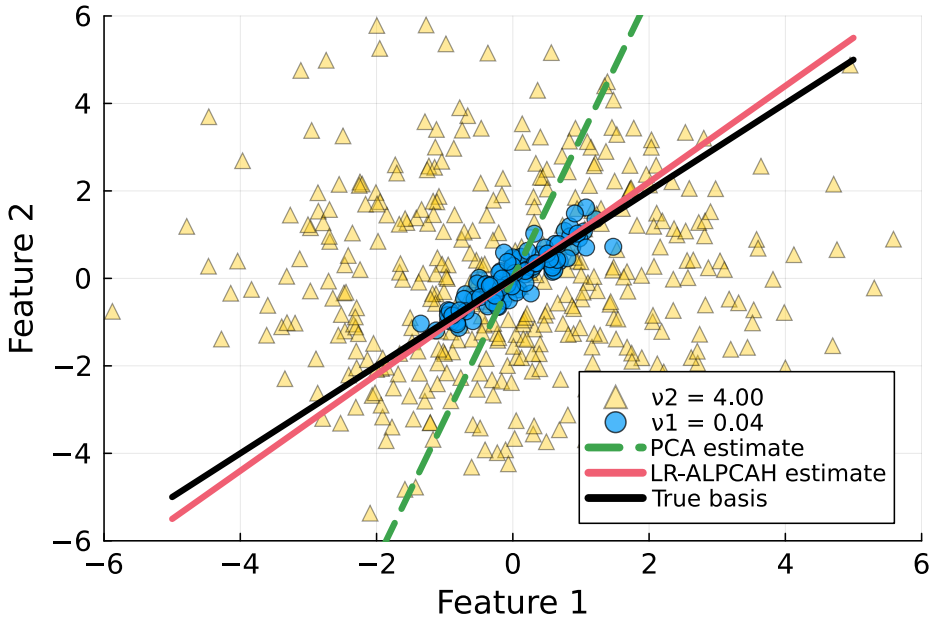


Figure 3.1: 1D subspace with data consisting of two noise groups shown with circle and triangle markers.

3.1 Introduction

Many modern data-science problems require learning an approximate signal subspace basis for some collection of data. This process is important for downstream tasks involving the subspace

The work in this chapter first appeared in conference proceedings [12] and was later published in IEEE Transactions on Signal Processing (TSP) [13].

basis coefficients, such as classification [56], regression [57], and compression [58]. More concretely, lesion detection [59], motion estimation [60], dynamic MRI reconstruction [61], and image/video denoising [62] are practical applications involving the estimation of a subspace basis. In the modern "big data" world, a significant amount of data is collected to solve problems, and this data tends to belong to a high-dimensional ambient space. However, the underlying relationships between the data features are often low-dimensional so the problem shifts towards finding low-dimensional structure in the data.

Some applications involve heterogeneous data that vary in quality due in part to noise characteristics associated with each data sample. Some examples of heteroscedastic data include environmental air data [63], astronomical spectral data [64], and biological sequencing data [65]. In heteroscedastic settings, the noisier data samples can significantly corrupt conventional basis estimates [66]. Subspace learning methods like probabilistic PCA (PPCA) [67] work well in the homoscedastic setting, meaning when the data is of the same quality throughout, but fail to accurately estimate bases in the heteroscedastic setting [68]. This limitation is due to implicit assumptions such as assuming that each sample's noise distribution is the same throughout (PPCA), or in the case of the classical Robust PCA (RPCA) method [69], that there are fewer outliers than good quality data samples.

A natural approach could be to simply discard the noisiest samples to avoid this issue. This approach requires the user to know the data quality, which may be unavailable in practice. That approach also assumes that there is enough good data to estimate the basis, but it is possible that a lack of good data requires using the noisy data, especially if the subspace dimension is higher than the number of good data points. Furthermore, even the noisier samples can help improve the basis estimate if properly modeled [68], so it is preferable to use all available data. This chapter introduces subspace learning algorithms that can estimate the sample-wise noise variances and use this information in the model to improve the estimate of the subspace basis associated with the low-rank structure of the data. See Fig. 3.1 for a visualization where PCA fails to account for heteroscedasticity in a simple 2D data example, but our LR-ALPCAH method more accurately finds the subspace basis.

The proposed subspace learning method, ALPCAH, was first introduced in previous proceedings work [12], allows for the optional use of rank knowledge via a low-rank promoting functional and makes no distributional assumptions about the low-rank component of the data, allowing it to achieve higher accuracy than current methods without knowing the noise variances. Moreover, we extend our previous proceedings work [12] by developing an alternative formulation inspired by the matrix factorization literature [70], that saves both memory and computing time at the cost of requiring the subspace dimension to be known or estimated.

The chapter is organized as follows. Section 3.2 introduces the heteroscedastic problem formulation for subspace learning and discusses related work. Section 3.3 introduces two subspace learning methods, one with nuclear norm style low-rank regularization originally introduced in our proceedings paper [12], and an extension to a regularization-free maximum likelihood approach. Section 3.4 covers synthetic and real data experiments that illustrate the effectiveness of these methods. Finally, Section 3.5 discusses some limitations of our methods and possible extensions.

3.2 Problem Formulation & Related Works

Let $\boldsymbol{y}_i \in \mathbb{R}^D$ denote the data samples for index $i \in \{1, \ldots, N\}$ given N total samples, and let D denote the ambient dimension. Let $\boldsymbol{x}_i$ represent the low-dimensional data sample generated by $\boldsymbol{x}_i = \boldsymbol{U}\boldsymbol{z}_i$ where $\boldsymbol{U} \in \mathbb{R}^{D\times d}$ is an unknown subspace basis of dimension d and $\boldsymbol{z}_i \in \mathbb{R}^d$ are the corresponding basis coordinates. Collect the measurements into a matrix $\boldsymbol{Y} = [\, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_N \,]$. Then, the heteroscedastic model we consider is

$$\boldsymbol{y}_i = \boldsymbol{x}_i + \boldsymbol{\epsilon}_i \quad \text{where} \quad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\boldsymbol{0}, \nu_i \boldsymbol{I}) \tag{3.1}$$

assuming Gaussian noise with variance ν_i, where $\boldsymbol{I}$ denotes the $D \times D$ identity matrix. We consider both the case where each data point may have its own noise variance, and cases where there are G groups of data having shared noise variance terms $\{\nu_1, \ldots, \nu_G\}$. Sec. 3.3 proposes an optimization problem that estimates the heterogeneous noise variances $\{\nu_i\}$ and the subspace basis $\boldsymbol{U}$.

3.2.1 Heteroscedastic Impact on Subspace Quality

Before describing the methods, we illustrate how heteroscedastic data impacts the quality of the PCA subspace basis estimate $\hat{\boldsymbol{U}}_{:,1:d}$, the first d columns of $\hat{\boldsymbol{U}}$. Let $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{E}$ where $\boldsymbol{X} \in \mathbb{R}^{D\times N}$ is a rank-d matrix and $\boldsymbol{E} \in \mathbb{R}^{D\times N}$ is the noise matrix where $\boldsymbol{E}_{:,i} \sim \mathcal{N}(\boldsymbol{0}, \nu_i \boldsymbol{I})\ \ \forall j$. Let $\boldsymbol{Y} = \hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{V}}'$ and $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}'$ denote singular value decompositions of their respective matrices, $\sigma_i(\boldsymbol{A})$ denotes the ith singular value of $\boldsymbol{A}$. Let $\|\boldsymbol{A}\|_2$ denote the spectral norm of matrix $\boldsymbol{A}$ and $\|\boldsymbol{x}\|_2$ denote the Euclidean norm of a vector $\boldsymbol{x}$. The notation $a \lesssim b$ means $\exists k > 0$ s.t. $a \leq kb$. By Wedin-Davis-Kahan $\sin\theta$ theorem [71, p. 95], it is known that

$$\|\hat{\boldsymbol{U}}_{:,1:d}\hat{\boldsymbol{U}}'_{:,1:d} - \boldsymbol{U}_{:,1:d}\boldsymbol{U}'_{:,1:d}\|_2 \leq \frac{2\|\boldsymbol{E}\|_2}{\sigma_d(\boldsymbol{X}) - \sigma_{d+1}(\boldsymbol{X})}. \tag{3.2}$$

This inequality states that the maximum angle of misalignment between the latent subspace basis $\boldsymbol{U}_{:,1:d}$ and the SVD-estimated subspace $\hat{\boldsymbol{U}}_{:,1:d}$ is bounded by the spectral norm of the noise

matrix over the spectral gap in matrix $\boldsymbol{X}$. Assuming the elements in $\boldsymbol{E}$ are zero mean and independent (not necessarily identically distributed) random variables it is known from [72] that, in expectation, the spectral norm of $\boldsymbol{E}$ is bounded as

$$\mathbb{E}[\|\boldsymbol{E}\|_2] \lesssim \max_i \sqrt{\sum_j \mathbb{E}[E_{ij}^2]} + \max_j \sqrt{\sum_i \mathbb{E}[E_{ij}^2]} + \sqrt[4]{\sum_{i,j} \mathbb{E}[E_{ij}^4]}. \tag{3.3}$$

Because $\boldsymbol{E}_{:,i} \sim \mathcal{N}(\boldsymbol{0}, \nu_i \boldsymbol{I}) \ \forall j$ in our application, it can be verified that

$$\max_i \sqrt{\sum_j \mathbb{E}[E_{ij}^2]} = \sqrt{\nu_{\text{sum}}^{(1)}} \tag{3.4}$$

$$\max_j \sqrt{\sum_i \mathbb{E}[E_{ij}^2]} = \sqrt{D\nu_{\max}} \tag{3.5}$$

$$\sqrt[4]{\sum_{i,j} \mathbb{E}[E_{ij}^4]} = \sqrt[4]{3D\nu_{\text{sum}}^{(2)}} \tag{3.6}$$

for $\nu_{\max} = \max_i \nu_i$ and $\nu_{\text{sum}}^{(k)} = \sum_i \nu_i^k$. Let $\boldsymbol{C_X}$ correspond to the covariance matrix of $\boldsymbol{X}$, i.e., $\boldsymbol{C_X} = \frac{1}{N}\boldsymbol{X}\boldsymbol{X}'$. Combining these bounds with the property that $\sigma_{d+1}(\boldsymbol{X}) = 0$ for a rank-d matrix leads to the following result. The subspace error, or more precisely, the maximum angle separation between the true subspace basis $\boldsymbol{U}_{:,1:d}$ and the estimated subspace basis $\hat{\boldsymbol{U}}_{:,1:d}$ is bounded as follows

$$\mathbb{E}[\|\hat{\boldsymbol{U}}_{:,1:d}\hat{\boldsymbol{U}}'_{:,1:d} - \boldsymbol{U}_{:,1:d}\boldsymbol{U}'_{:,1:d}\|_2^2] \lesssim \frac{\left(\sqrt{\nu_{\text{sum}}^{(1)}} + \sqrt{D\nu_{\max}} + \sqrt[4]{3D\nu_{\text{sum}}^{(2)}}\right)^2}{N\sigma_d(C_X)}. \tag{3.7}$$

This upper bound indicates that the quality of the subspace basis estimate $\hat{\boldsymbol{U}}_{:,1:d}$ provided by the SVD of noisy data $\boldsymbol{Y}$, i.e., by conventional PCA, could be degraded by heteroscedastic noise. Fig. 3.8 in the appendix provides empirical evidence for this claim. Thus, it can be advantageous to model the heteroscedasticity and design a more robust PCA-like algorithm that mitigates some of the effects of heteroscedastic noise and achieves more accurate estimates of the subspace basis.

3.2.2 Other Heteroscedastic Models

This chapter focuses on heteroscedastic noise across the data samples. There are other methods in the literature that explore heteroscedasticity in different ways. For example, HeteroPCA considers heteroscedasticity across the feature space [73]. One possible application of that model is for data that consists of sensor information with multiple devices that naturally have different levels of precision and signal-to-noise ratio (SNR). Another heterogeneity model considers the noise to be homoscedastic and instead assumes that the signal itself is heteroscedastic [74]. In that case, the power fluctuating signals are embedded in white Gaussian noise. Each of these models has its own family of applications.

3.2.3 Probabilistic PCA (PPCA)

PCA methods like PPCA [67] work well in the homoscedastic setting, i.e., when the data is of the same quality throughout, but fail to accurately estimate the basis when the data varies in quality, e.g., in the heteroscedastic setting [12].

Let $\boldsymbol{C} = \boldsymbol{F}\boldsymbol{F}' + \nu\boldsymbol{I}$ and observe that the model

$$\boldsymbol{y}_i = \boldsymbol{F}\boldsymbol{z}_i + \boldsymbol{\epsilon} \tag{3.8}$$

$$\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}),\ \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \nu\boldsymbol{I}),\ \boldsymbol{y}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}) \tag{3.9}$$

is similar to (3.1) in that we have observation data $\boldsymbol{y}_i$, unobserved variables $\boldsymbol{z}_i$, factor matrix $\boldsymbol{F}$, and noise term $\boldsymbol{\epsilon}$. Then, for a covariance-type matrix $\boldsymbol{C}_y = \sum_i \boldsymbol{y}_i\boldsymbol{y}_i'$ formed from data samples $\boldsymbol{Y}$, the negative log-likelihood is

$$\mathcal{L}(\boldsymbol{F}, \nu) = -\frac{N}{2}(d\log(2\pi) + \log(|\boldsymbol{C}|) + \mathrm{Tr}(\boldsymbol{C}^{-1}\boldsymbol{C}_y)), \tag{3.10}$$

where $|\cdot|$ and $\mathrm{Tr}(\cdot)$ denote matrix determinant and trace, respectively. After estimating $\boldsymbol{F}$ and ν by minimizing (3.10), PPCA finds the subspace basis by orthogonalizing $\boldsymbol{F}$. Because $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \nu\boldsymbol{I})$ is identically distributed across all data samples, PPCA does not account for heterogeneous data samples.

3.2.4 Robust PCA (RPCA)

Robust PCA (RPCA) [69] decomposes the data matrix $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{E}$ into a low-rank component $\boldsymbol{X}$ and an outlier matrix $\boldsymbol{E}$ by the following optimization problem:

$$\underset{\boldsymbol{X},\boldsymbol{E}}{\arg\min} \left(\lambda\|\boldsymbol{X}\|_* + \|\boldsymbol{E}\|_{1,1}\right) \text{ s.t. } \boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{E} \tag{3.11}$$

where $\|\boldsymbol{X}\|_* = \sum_{i=1} \sigma_i(\boldsymbol{X})$ and $\|\boldsymbol{E}\|_{1,1} = \sum_{i,j} |E_{ij}|$. RPCA finds the subspace basis by iteratively applying an SVD to $\boldsymbol{X}$ to soft threshold the singular values. Here, the term $\|\boldsymbol{E}\|_{1,1}$ encourages sparsity and so captures noise in the data matrix by assuming there is a sparse collection of outliers. This modeling assumption may not be true in some applications. For instance, low-quality and abundant commercial sensors are often combined with fewer high-quality sensors. Ref. [12] illustrated the limitations of RPCA in the heteroscedastic regime.

3.2.5 Weighted PCA (WPCA)

Given data samples $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\}$ and weights $\{w_1, \ldots, w_N\}$, the weighted PCA (WPCA) approach [75] for modeling heteroscedastic data forms the following weighted sample covariance matrix

$$\boldsymbol{C}_y(\boldsymbol{w}) = \sum_{i=1}^{N} w_i(\boldsymbol{y}_i \boldsymbol{y}_i^T), \tag{3.12}$$

where a natural choice for the weights is $w_i = \nu_i^{-1}$. WPCA finds the subspace basis by orthogonalizing $\boldsymbol{C}_y(\boldsymbol{w})$ via eigenvalue decomposition (EVD). However, the noise variances may not be known, e.g., unknown dataset origin or unavailable data sheet for physical sensors.

3.2.6 Heteroscedastic PPCA Technique (HePPCAT)

To our knowledge, besides ALPCAH, there is only one sample-based heteroscedastic PCA algorithm that estimates unknown noise variances. The Heteroscedastic Probabilistic PCA Technique (HePPCAT) [68] builds on the PPCA formulation. For $n_1 + \ldots + n_G = N$ data samples from G noise groups, the model is described as

$$\boldsymbol{y}_{g,i} = \boldsymbol{F}\boldsymbol{z}_{g,i} + \boldsymbol{\epsilon}_{g,i} \quad i \in \{1, \ldots, n_G\},\ g \in \{1, \ldots, G\} \tag{3.13}$$

for factor scores $\boldsymbol{z}_{g,i} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, noise terms $\boldsymbol{\epsilon}_{g,i} \sim \mathcal{N}(\boldsymbol{0}, v_g\boldsymbol{I})$, and points $\boldsymbol{y}_{g,i} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}_g)$ where $\boldsymbol{C}_g = \boldsymbol{F}\boldsymbol{F}' + v_g\boldsymbol{I}$ for factor matrix $\boldsymbol{F}$. Then, the negative log-likelihood model to optimize is the

following

$$\mathcal{L}(\boldsymbol{F}, \boldsymbol{v}) = \frac{1}{2}\sum_{g=1}^{G}[n_g \ln \det(\boldsymbol{C}_g)^{-1} - \text{Tr}\{\boldsymbol{Y}_{(g)}^T(\boldsymbol{C}_g)^{-1}\boldsymbol{Y}_{(g)}\}] \tag{3.14}$$

where $\boldsymbol{Y}_{(g)}$ denotes the submatrix of $\boldsymbol{Y}$ that consists only of data samples belonging to the gth noise group, and $\boldsymbol{v} = (v_1, \ldots, v_G)$ denotes the unknown nose variances for each group. Being a factor analysis method, HePPCAT makes Gaussian assumptions about the basis coefficients $\boldsymbol{z}_{l,i}$ that may not be a good model for some datasets. Additionally, HePPCAT requires the rank parameter d associated with the latent signal matrix $\boldsymbol{X}$ to be estimated or known a priori.

3.3 Proposed Subspace Learning Methods

This section introduces the ALPCAH formulation for subspace learning. Since nuclear norm computation is expensive for big data applications due to SVD computations, we take inspiration from the matrix factorization literature and additionally develop LR-ALPCAH to be a fast and memory-efficient alternative to ALPCAH.

3.3.1 ALPCAH

For the measurement model $\boldsymbol{y}_i \sim \mathcal{N}(\boldsymbol{x}_i, \nu_i \boldsymbol{I})$ in (3.1), the probability density function for a single data vector $\boldsymbol{y}_i$ is

$$\frac{1}{\sqrt{(2\pi)^D |\nu_i \boldsymbol{I}|}} \exp\left[-\frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{x}_i)'(\nu_i \boldsymbol{I})^{-1}(\boldsymbol{y}_i - \boldsymbol{x}_i)\right]. \tag{3.15}$$

For uncorrelated samples, after dropping constants, the joint log likelihood of all data $\{\boldsymbol{y}_i\}_{i=1}^N$ is the following

$$\sum_{i=1}^{N} -\frac{1}{2}\log|\nu_i \boldsymbol{I}| - \frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{x}_i)'(\nu_i \boldsymbol{I})^{-1}(\boldsymbol{y}_i - \boldsymbol{x}_i). \tag{3.16}$$

Let $\boldsymbol{\Pi} = \text{diag}(\nu_1, \ldots, \nu_N) \in \mathbb{R}^{N \times N}$ be a diagonal matrix representing the (typically unknown) noise variances. Then, the negative log likelihood in matrix form is

$$\begin{aligned} &\frac{D}{2}\log|\boldsymbol{\Pi}| + \frac{1}{2}\text{Tr}[(\boldsymbol{Y} - \boldsymbol{X})\boldsymbol{\Pi}^{-1}(\boldsymbol{Y} - \boldsymbol{X})'] \\ =&\frac{D}{2}\log|\boldsymbol{\Pi}| + \frac{1}{2}\|(\boldsymbol{Y} - \boldsymbol{X})\boldsymbol{\Pi}^{-1/2}\|_{\text{F}}^2, \end{aligned} \tag{3.17}$$

using trace lemmas. When both $\boldsymbol{\Pi}$ and $\boldsymbol{X}$ are unknown, pursuing maximum-likelihood estimation with (3.17) would lead to degenerate solutions. Thus, regularization is necessary to promote a

low-rank solution. In this work, we use a functional modified from the nuclear norm regularizer to encourage the estimate of $\boldsymbol{X}$ to be low-rank.

The optimization problem used by ALPCAH for the heteroscedastic model is

$$\underset{\boldsymbol{X},\boldsymbol{\Pi}}{\arg\min}\, \lambda f_{\hat{d}}(\boldsymbol{X}) + \frac{1}{2}\|(\boldsymbol{Y}-\boldsymbol{X})\boldsymbol{\Pi}^{-1/2}\|_{\mathrm{F}}^2 + \frac{D}{2}\log|\boldsymbol{\Pi}|, \tag{3.18}$$

where $f_{\hat{d}}(\boldsymbol{X})$ is a novel functional [76] that promotes low-rank structure in $\boldsymbol{X}$, $\hat{d}$ is the rank parameter, and $\lambda \in \mathbb{R}^+$ is a regularization parameter. In the following, we introduce our algorithm called ALPCAH (**A**lgorithm for **L**ow-rank regularized **PCA** for **H**eteroscedastic data) for solving (3.18). Since $\boldsymbol{X}$ represents the denoised data matrix, the subspace basis is calculated by SVD on the optimal solution from (3.18) and extracting the first $\hat{d}$ left singular vectors so that $\hat{\boldsymbol{X}} = \sum_i \hat{\sigma}_i \hat{\boldsymbol{u}}_i \hat{\boldsymbol{v}}_i'$ and thus $\hat{\boldsymbol{U}} = [\hat{\boldsymbol{u}}_1, \ldots, \hat{\boldsymbol{u}}_{\hat{d}}]$. The low-rank promoting functional we use is the sum of the tail singular values defined as

$$f_{\hat{d}}(\boldsymbol{X}) \triangleq \sum_{i=\hat{d}+1}^{\min(D,N)} \sigma_i(\boldsymbol{X}) = \|\boldsymbol{X}\|_* - \|\boldsymbol{X}\|_{\mathrm{Ky-Fan}(\hat{d})} \tag{3.19}$$

where $\|\cdot\|_*$ denotes the nuclear norm, and $\|\cdot\|_{\mathrm{Ky-Fan}(\hat{d})}$ denotes the Ky-Fan norm [77] defined as the sum of the first $\hat{d}$ singular values. For $\hat{d}=0$, $f_0(\boldsymbol{X}) = \|\boldsymbol{X}\|_*$. For a general $\hat{d} > 0$, $f_{\hat{d}}(\boldsymbol{X})$ is a nonconvex difference of convex functions. We use the functional $f_{\hat{d}}$ instead of the nuclear norm since we empirically found that the nuclear norm tends to over-shrink the singular values of $\boldsymbol{X}$ in the heteroscedastic setting. Here, the rank parameter $\hat{d} \ll D$ is either known beforehand, estimated using methods like row permutations [78] or sign flips [79], or intentionally over-parameterized.

Definition 1. *Let $\boldsymbol{A} \in \mathbb{R}^{D\times N}$ be a rank k matrix such that its decomposition is $\mathrm{SVD}(\boldsymbol{A}) = \boldsymbol{U}_A\boldsymbol{D}_A\boldsymbol{V}_A'$ where $\boldsymbol{D}_A = \mathrm{diag}(\sigma_1(\boldsymbol{A}), \ldots, \sigma_{\min(D,N)}(\boldsymbol{A}))$. Let the soft thresholding operation be defined as $\mathcal{S}_\tau[x] = \mathrm{sign}(x)\max(|x|-\tau, 0)$ for some threshold $\tau > 0$. Decompose $\boldsymbol{D}_A$ such that $\boldsymbol{D}_A = \boldsymbol{D}_{A1} + \boldsymbol{D}_{A2} = \mathrm{diag}(\sigma_1(\boldsymbol{A}), \ldots, \sigma_{\hat{d}}(\boldsymbol{A}), 0, \ldots, 0) +$
$\mathrm{diag}(0, \ldots, 0, \sigma_{\hat{d}+1}(\boldsymbol{A}), \ldots, \sigma_N(\boldsymbol{A}))$. Then, the proximal map for $f_{\hat{d}}$ is the tail singular value thresholding operation [76]:*

$$\mathrm{TSVT}(\boldsymbol{A}, \tau, \hat{d}) \triangleq \boldsymbol{U}_A\left(\boldsymbol{D}_{A1} + \mathcal{S}_\tau[\boldsymbol{D}_{A2}]\right)\boldsymbol{V}_A'. \tag{3.20}$$

Although the proximal operator for $f_{\hat{d}}$ is provided in (3.20), it is unclear how one would apply a proximal gradient method (PGM) directly to (3.18) due to the product of $\boldsymbol{X}$ and $\boldsymbol{\Pi}$. One could apply a block coordinate descent approach that alternates between updating $\boldsymbol{X}$ using a PGM, and updates the diagonal elements of $\boldsymbol{\Pi}$ using a closed-form solution. The PGM update of $\boldsymbol{X}$ could cause slow convergence because the Lipschitz constant of the gradient of the smooth term $g(\boldsymbol{X})$ is the reciprocal of the smallest diagonal element of $\boldsymbol{\Pi}$, which could be quite large, leading to small

step sizes. Thus, instead, we optimize (3.18) using the inexact augmented Lagrangian method known as the alternating direction method of multipliers (ADMM) [80] that introduces auxiliary variables to convert a complicated optimization problem into a sequence of simpler optimization problems.

Defining the auxiliary variable $\boldsymbol{Z} = \boldsymbol{Y} - \boldsymbol{X}$, the augmented penalty parameter $\mu \in \mathbb{R}$, and dual variable $\boldsymbol{\Lambda} \in \mathbb{R}^{D \times N}$, the augmented Lagrangian, as defined in [81], is

$$\begin{aligned}\mathcal{L}_\mu(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\Lambda}, \boldsymbol{\Pi}) = \lambda f_{\hat{d}}(\boldsymbol{X}) + \frac{1}{2}\|\boldsymbol{Z}\boldsymbol{\Pi}^{-1/2}\|_{\mathrm{F}}^2 + \frac{D}{2}\log|\boldsymbol{\Pi}| \\ + \langle \boldsymbol{\Lambda}, \boldsymbol{Y} - \boldsymbol{X} - \boldsymbol{Z} \rangle + \frac{\mu}{2}\|\boldsymbol{Y} - \boldsymbol{X} - \boldsymbol{Z}\|_{\mathrm{F}}^2,\end{aligned} \tag{3.21}$$

where $\langle \cdot, \cdot \rangle$ denotes the Frobenius inner product between two matrices.

Performing a block Gauss-Seidel pass for each variable in (3.21) results in the following closed-form updates

$$\begin{aligned}\boldsymbol{Z}_{t+1} &= \arg\min_{\boldsymbol{Z}} \mathcal{L}_\mu(\boldsymbol{X}_t, \boldsymbol{Z}, \boldsymbol{\Lambda}_t, \boldsymbol{\Pi}_t) \\ &= \left[\mu\left(\boldsymbol{Y} - \boldsymbol{X}_t\right) + \boldsymbol{\Lambda}_t\right]\left(\boldsymbol{\Pi}_t^{-1} + \mu\boldsymbol{I}\right)^{-1} \qquad (3.22) \\ \boldsymbol{X}_{t+1} &= \arg\min_{\boldsymbol{X}} \mathcal{L}_\mu(\boldsymbol{X}, \boldsymbol{Z}_t, \boldsymbol{\Lambda}_t, \boldsymbol{\Pi}_t) \\ &= \mathrm{TSVT}(\boldsymbol{Y} - \boldsymbol{Z}_t + \frac{1}{\mu}\boldsymbol{\Lambda}_t, \frac{\lambda}{\mu}, \hat{d}) \qquad (3.23) \\ \boldsymbol{\Lambda}_{t+1} &= \boldsymbol{\Lambda}_t + \mu\left(\boldsymbol{Y} - \boldsymbol{X}_t - \boldsymbol{Z}_t\right) \qquad (3.24)\end{aligned}$$

for current iteration pass t. Each pass is run for T total iterations. When we treat the data sample $\boldsymbol{y}_i$ as having its own unknown noise variance, then the variance update is

$$\boldsymbol{\Pi}_{t+1} = \arg\min_{\boldsymbol{\Pi}} \mathcal{L}_\mu(\boldsymbol{X}_t, \boldsymbol{Z}_t, \boldsymbol{\Lambda}_t, \boldsymbol{\Pi}) = \frac{1}{D}\boldsymbol{Z}_t'\boldsymbol{Z}_t \odot \boldsymbol{I}. \tag{3.25}$$

For the case when the data points have grouped noise variances, let $g \in \{1, \ldots, G\}$ signify the gth noise group out of G total groups with n_g denoting the number of samples in the gth group; then the grouped noise variance update instead becomes

$$\nu_g = \frac{1}{Dn_g}\|\boldsymbol{Z}_{(g)}\|_{\mathrm{F}}^2 = \frac{1}{Dn_g}\|\boldsymbol{Y}_{(g)} - \boldsymbol{X}_{(g)}\|_{\mathrm{F}}^2 \tag{3.26}$$

where the notation $\boldsymbol{Y}_{(g)}$ denotes the submatrix of $\boldsymbol{Y}$ that consists solely of data samples from the gth noise group.

3.3.1.1 Convergence with known variance

Consider the cost function for the case when the variances $\boldsymbol{\Pi}$ are known. The formulation consists of a two-block setup written as

$$\underset{\boldsymbol{X},\boldsymbol{Z}}{\arg\min}\, \lambda f_{\hat{d}}(\boldsymbol{X}) + \frac{1}{2}\|\boldsymbol{Z}\boldsymbol{\Pi}^{-1/2}\|_{\mathrm{F}}^2 \text{ s.t. } \boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{Z}. \tag{3.27}$$

Theorem 1. *Let* $\Psi(\boldsymbol{X},\boldsymbol{Z}) = f(\boldsymbol{X}) + g(\boldsymbol{Z})$ *where* $f(\boldsymbol{X}) = \lambda f_{\hat{d}}(\boldsymbol{X})$ *and* $g(\boldsymbol{Z}) = \frac{1}{2}\|\boldsymbol{Z}\boldsymbol{\Pi}^{-1/2}\|_{\mathrm{F}}^2$. *Let* $\nu_i \geq \epsilon > 0 \;\; \forall i$. *Assuming that* μ *in* (3.21) *satisfies* $\mu > 2L_g = 2\|\boldsymbol{\Pi}^{-1}\|_2$, *the sequence* $\{(\boldsymbol{X}_t, \boldsymbol{Z}_t, \boldsymbol{\Lambda}_t, \boldsymbol{\Pi})\}_{i=1}^T$ *generated by ADMM in (3.22) (3.23) (3.24) (3.25) converges to a KKT (Karush–Kuhn–Tucker) point of the augmented Lagrangian* $\mathcal{L}_\mu(\boldsymbol{X},\boldsymbol{Z},\boldsymbol{\Lambda},\boldsymbol{\Pi})$ *with fixed* $\boldsymbol{\Pi}$.

Proof. ADMM convergence for nonconvex problems has been explored for two-block setups [82]. The functional $f(\boldsymbol{X})$ is a proper, lower semi-continuous function since it is a sum of continuous functions. The function $g(\boldsymbol{Z})$ is a continuous differentiable function whose gradient is Lipschitz continuous with modulus of continuity $L_g = \|\boldsymbol{\Pi}^{-1}\|_2$. By definition $g(\boldsymbol{Z}) = \nu_1^{-1/2}Z_{11} + \nu_1^{-1/2}Z_{21} + \ldots + \nu_N^{-1/2}Z_{DN}$. Since $g(\boldsymbol{Z})$ is a polynomial equation, its graph is a semi-algebraic set.

Let $\boldsymbol{G} = \boldsymbol{X}'\boldsymbol{X} \in \mathbb{R}^{N\times N}$. Then, by Cayley Hamilton theorem, the characteristic polynomial of $\boldsymbol{G}$ is $p_G(z) = z^N + c_{n-1}(\boldsymbol{G})z^{N-1} + \ldots + c_1(\boldsymbol{G})z + c_0$ for constants $c_i \in \mathbb{R}$ and polynomial degree N. Let λ denote an eigenvalue of $\boldsymbol{G}$ which implies $p_G(\lambda) = 0$. Then, the set $\mathcal{S}_G = \{\lambda \mid p_G(\lambda) = 0\}$ is semi-algebraic since it is defined by polynomial equations. Note that $\lambda_i(\boldsymbol{G}) = \sigma_i^2(\boldsymbol{X})$ since $\boldsymbol{G}$ is the Gram matrix of $\boldsymbol{X}$. The set $\mathcal{S}_X = \{\sigma \mid \sigma^2 = \lambda \in S_G, \sigma \geq 0\} = \{\sigma_1, \ldots, \sigma_N\}$ is semi-algebraic as it is expressed in terms of polynomial inequalities. Expressing $h(\boldsymbol{X}) = \|\boldsymbol{X}\|_* = h(\sigma_1, \ldots, \sigma_N)$, its graph $h = \{(\sigma, f(\sigma))\}$ is semi-algebraic and thus by extension so is the nuclear norm.

By Tarksi-Seidenburg theorem [83, p. 345], defining the map $\Phi : \mathbb{R}^n \to \mathbb{R}^{\hat{d}}$ that retains the first $\hat{d}$ singular values of $\mathcal{S}_X$, the set $\Phi(\mathcal{S}_X) = \{\sigma_1, \ldots, \sigma_{\hat{d}}\}$ is semi-algebraic and thus so is $q(\boldsymbol{X}) = \|\boldsymbol{X}\|_{\mathrm{Ky-Fan}(\hat{d})}$. A finite weighted sum of semi-algebraic functions is known to be semi-algebraic [84] and so $f(\boldsymbol{X}) = h(\boldsymbol{X}) - q(\boldsymbol{X})$ is semi-algebraic. Since the functions $f(\boldsymbol{X})$ and $g(\boldsymbol{Z})$ are lower, semi-continuous and definable on an o-minimal structure such as semi-algebraic [85], it follows that $\Psi(\boldsymbol{X},\boldsymbol{Z}) = f(\boldsymbol{X}) + g(\boldsymbol{Z})$ is a Kurdyka-Lojasiewicz function [84]. Thus the sequence $\{(\boldsymbol{X}_t, \boldsymbol{Z}_t, \boldsymbol{\Lambda}_t, \boldsymbol{\Pi})\}_{i\in\mathbb{N}}$ converges to a KKT point by [82, Thm. 3.1]. □

3.3.2 LR-ALPCAH

The main computational expense for the ALPCAH algorithm is the SVD operations used in every iteration of complexity $\mathcal{O}(DN\min(D,N))$. To reduce computation, we take inspiration from

the matrix factorization literature [70] and factorize $\boldsymbol{X} \in \mathbb{R}^{D\times N} \approx \boldsymbol{L}\boldsymbol{R}'$ where $\boldsymbol{L} \in \mathbb{R}^{D\times \hat{d}}$ and $\boldsymbol{R} \in \mathbb{R}^{N\times \hat{d}}$ for some rank estimate $\hat{d}$. Using the factorized form, we propose to estimate $\boldsymbol{X}$ by solving for $\boldsymbol{L}$ and $\boldsymbol{R}$ in the following optimization problem

$$\begin{aligned}\hat{\boldsymbol{L}}, \hat{\boldsymbol{R}}, \hat{\boldsymbol{\Pi}} &= \underset{\boldsymbol{L},\boldsymbol{R},\boldsymbol{\Pi}}{\arg\min}\ f(\boldsymbol{L}, \boldsymbol{R}, \boldsymbol{\Pi}) \\ f(\boldsymbol{L}, \boldsymbol{R}, \boldsymbol{\Pi}) &= \frac{1}{2}\|(\boldsymbol{Y} - \boldsymbol{L}\boldsymbol{R}')\boldsymbol{\Pi}^{-1/2}\|_{\mathrm{F}}^2 + \frac{D}{2}\log|\boldsymbol{\Pi}|. \end{aligned} \tag{3.28}$$

This version is a maximum-likelihood estimator of $\boldsymbol{\Pi}$ and the factors $\boldsymbol{L}$ and $\boldsymbol{R}$. This comes from a modified model (3.1) where

$$\boldsymbol{y}_i = \boldsymbol{L}\boldsymbol{r}_i + \boldsymbol{\epsilon}_i, \quad \boldsymbol{\epsilon}_i \sim N(\boldsymbol{0}, \nu_i \boldsymbol{I}), \tag{3.29}$$

where $\boldsymbol{r}_i$ denotes the ith column of $\boldsymbol{R}$. We call this version LR-ALPCAH given the prevalence of $\boldsymbol{L}\boldsymbol{R}'$ notation in the matrix factorization literature. The crucial difference between ALPCAH and LR-ALPCAH is that ALPCAH uses a "soft" low rank constraint through the regularization penalty λ with optional usage of $\hat{d}$, whereas LR-ALPCAH uses a "hard" low rank constraint since $\boldsymbol{L}$ and $\boldsymbol{R}$ rigidly contain $\hat{d}$ columns.

We solve this optimization problem using alternating minimization [86] to solve each sub-block, resulting in the following updates:

$$\begin{aligned}\boldsymbol{L}_{t+1} &= \underset{\boldsymbol{L}}{\arg\min}\, f(\boldsymbol{L}, \boldsymbol{R}_t, \boldsymbol{\Pi}_t) \\ &= \boldsymbol{Y}\boldsymbol{\Pi}_t^{-1}\boldsymbol{R}_t(\boldsymbol{R}_t'\boldsymbol{\Pi}_t^{-1}\boldsymbol{R}_t)^{-1} \qquad (3.30) \\ \boldsymbol{R}_{t+1} &= \underset{\boldsymbol{R}}{\arg\min}\, f(\boldsymbol{L}_t, \boldsymbol{R}, \boldsymbol{\Pi}_t) \\ &= \boldsymbol{Y}'\boldsymbol{L}_t(\boldsymbol{L}_t'\boldsymbol{L}_t)^{-1} \qquad (3.31) \\ \boldsymbol{\Pi}_{t+1} &= \underset{\boldsymbol{\Pi}}{\arg\min}\, f(\boldsymbol{L}_t, \boldsymbol{R}_t, \boldsymbol{\Pi}) \implies \\ \boldsymbol{e}_j'\boldsymbol{\Pi}_{t+1}\boldsymbol{e}_j &= D^{-1}\|(\boldsymbol{Y} - \boldsymbol{L}_t\boldsymbol{R}_t')\boldsymbol{e}_j\|_2^2,\ \forall j, \qquad (3.32)\end{aligned}$$

where $\boldsymbol{e}_j$ denotes the jth standard canonical basis vector that we use to select the jth column of some matrix. The $\boldsymbol{\Pi}_t$ update (3.32) is the same as (3.25) in that each point is treated as having its own noise variance and both equations perform the same operation. This implementation requires less computation and memory since the matrix $\boldsymbol{Z}_t'\boldsymbol{Z}_t$ is not formed. One can substitute (3.32) with (3.26) if noise grouping is known.

Since this is a nonconvex problem, initialization will play a key role in the success of optimization. First, we initialize the $\boldsymbol{L}_t$ and $\boldsymbol{R}_t$ matrices with the following spectral approach:

$$\begin{aligned}\text{Spectral Init}(\boldsymbol{Y}) = \hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{V}}' &\approx (\hat{\boldsymbol{U}}_{1:\hat{d}}\hat{\boldsymbol{\Sigma}}_{1:\hat{d}}^{1/2})(\hat{\boldsymbol{\Sigma}}_{1:\hat{d}}^{1/2}\hat{\boldsymbol{V}}_{1:\hat{d}}') \\ &= (\boldsymbol{L}_0)(\boldsymbol{R}_0'). \end{aligned} \tag{3.33}$$

This initialization from the matrix factorization literature [70] [87] is a natural approach due to the Eckart-Young theorem [88] that shows $\boldsymbol{L}_0$ and $\boldsymbol{R}_0$ are the best rank-constrained matrices that solve

$$\arg\min_{\boldsymbol{L},\boldsymbol{R}} \|\boldsymbol{Y} - \boldsymbol{L}\boldsymbol{R}'\|_{\text{F}} \text{ subject to } \text{rank}(\boldsymbol{L}), \text{rank}(\boldsymbol{R}) \leq \hat{d}. \tag{3.34}$$

Second, we initialize the noise variances using the Euclidean norms of the columns of the residual $\boldsymbol{Y} - \boldsymbol{L}_0\boldsymbol{R}_0'$ with (3.32). Finally, we apply alternating minimization to update $\boldsymbol{L}_t$ and $\boldsymbol{R}_t$ matrices at current iteration t via (3.30) (3.31) (3.32).

Since $\boldsymbol{L}_t$ is not semi-unitary but has the same range as $\boldsymbol{U}$, we apply Gram-Schmidt orthogonalization to the final $\boldsymbol{L}_t$ matrix to estimate the subspace basis, as described in Alg. LR-ALPCAH. The matrix inversions used in the $\boldsymbol{L}_t$ and $\boldsymbol{R}_t$ updates involve $d \times d$ matrices that are relatively small and thus computationally feasible for many practical problems given complexity $\mathcal{O}(\hat{d}^3)$ knowing that $\hat{d} \ll \min(D, N)$. Combining the matrix multiplications and inversions, LR-ALPCAH has a per-iteration complexity of $\mathcal{O}(DN\hat{d} + \hat{d}^3)$. This is in contrast to ALPCAH with per-iteration complexity $\mathcal{O}(DN \min(D, N))$ due to the SVD computations.

3.3.2.1 Convergence with unknown variance

Note that (3.28) is a nonconvex function and we apply alternating minimization, also known as block coordinate descent or block nonlinear Gauss-Seidel method, to solve the optimization problem. Given a noise variance lower bound $\epsilon > 0$, the feasible sets for $\boldsymbol{L}, \boldsymbol{R}, \boldsymbol{\Pi}$ variables are given by

$$\mathcal{S}_L = \mathbb{R}^{D\times\hat{d}}, \quad \mathcal{S}_R = \mathbb{R}^{N\times\hat{d}} \tag{3.35}$$

$$\mathcal{S}_{\boldsymbol{\Pi}} = \{\boldsymbol{\Pi}_{i,j} \in [\epsilon, \infty) \ \ \forall i = j, \ 0 \text{ o.w.}\}. \tag{3.36}$$

Given the following optimization problem with

$$\begin{aligned} &\arg\min f(\boldsymbol{L}, \boldsymbol{R}, \boldsymbol{\Pi}) \\ &\text{subject to } \boldsymbol{L}, \boldsymbol{R}, \boldsymbol{\Pi} \in \mathcal{S} = \mathcal{S}_{\boldsymbol{L}} \times \mathcal{S}_{\boldsymbol{R}} \times \mathcal{S}_{\boldsymbol{\Pi}}, \end{aligned} \tag{3.37}$$

Algorithm LR-ALPCAH (github.com/javiersc1/ALPCAH)
(unknown variances, unknown quality noise grouping)

Input: $\boldsymbol{Y} \in \mathbb{R}^{D\times N}$: data, $\hat{d} \in \mathbb{N}^*$: rank estimate
Opt: $T \in \mathbb{N}^*$: iterations, $\epsilon \in \mathbb{R}^+$: variance noise floor
Output: $\boldsymbol{U} \in \mathbb{R}^{D\times \hat{d}}$: subspace basis, $\boldsymbol{X} \in \mathbb{R}^{D\times N}$: low-rank estimated data, $\boldsymbol{\nu} \in \mathbb{R}^+$: estimated noise variances
// sample mean to de-mean data
$\boldsymbol{\mu} \leftarrow \frac{1}{N}\boldsymbol{Y}\boldsymbol{1}$
// the method assumes linear subspaces only
$\boldsymbol{Y} \leftarrow \boldsymbol{Y} - \boldsymbol{1}'\boldsymbol{\mu}$
// initialize matrices by (3.33)
$\boldsymbol{L}_0, \boldsymbol{R}_0 \leftarrow$ SPECTRALINIT$(\boldsymbol{Y}, \hat{d})$
// compute noise variances from residuals $\boldsymbol{Y} - \boldsymbol{L}_0\boldsymbol{R}_0'$
// $\boldsymbol{e}_j$ is canonical basis vector
$\nu_0 \leftarrow \max_{j=1,\dots,N} \frac{1}{D}\|(\boldsymbol{Y} - \boldsymbol{L}_0\boldsymbol{R}_0')\boldsymbol{e}_j\|_2^2$
$\nu_0 \leftarrow \max(\nu_0, \epsilon)$
$\boldsymbol{\Pi}_0^{-1} \leftarrow (1/\nu_0)\boldsymbol{I}$
// update $\boldsymbol{L}, \boldsymbol{R}, \boldsymbol{\Pi}$ matrices using (3.30) (3.31) (3.32)
for $t = 1, \dots, T$ **do**
  $\boldsymbol{L}_t \leftarrow \boldsymbol{Y}\boldsymbol{\Pi}_{t-1}^{-1}\boldsymbol{R}_{t-1}(\boldsymbol{R}_{t-1}'\boldsymbol{\Pi}_{t-1}^{-1}\boldsymbol{R}_{t-1})^{-1}$
  $\boldsymbol{R}_t \leftarrow \boldsymbol{Y}_{t-1}'\boldsymbol{L}_{t-1}(\boldsymbol{L}_{t-1}'\boldsymbol{L}_{t-1})^{-1}$
  $\nu_j \leftarrow \max(\frac{1}{D}\|(\boldsymbol{Y} - \boldsymbol{L}_{t-1}\boldsymbol{R}_{t-1}')\boldsymbol{e}_j\|_2^2,\ \epsilon),\ j = 1, \dots, N$
  $\boldsymbol{\Pi}_t^{-1} \leftarrow$ Diagonal$(1/\nu_j)$
end for
// form subspace basis from final left factor
STATE $\boldsymbol{U} \leftarrow$ GRAMSCHMIDT$(\boldsymbol{L}_T)$
// construct de-meaned low-rank estimate
$\boldsymbol{X} \leftarrow \boldsymbol{L}_T\boldsymbol{R}_T'$
// add back original sample mean
$\boldsymbol{X} \leftarrow \boldsymbol{X} + \boldsymbol{1}'\boldsymbol{\mu}$

the following theorem establishes local convergence of $\{(\boldsymbol{L}_t, \boldsymbol{R}_t, \boldsymbol{\Pi}_t)\}_{t=1}^T$ to critical points of (3.37).

Theorem 2. *The sequence generated by alternating minimization $\{(\boldsymbol{L}_t, \boldsymbol{R}_t, \boldsymbol{\Pi}_t)\}_{t=1}^T$ in Alg. LR-ALPCAH has limit points that are critical points of (3.37).*

Proof. The cost function f is continuously differentiable by inspection of its terms. The feasible sets $\mathcal{S}_L, \mathcal{S}_R$ are trivially nonempty, closed, and convex sets by definition. Moreover, $\boldsymbol{\Pi} \in \mathcal{S}_{\boldsymbol{\Pi}}$ since it is an enforced constraint of the optimization in (3.37). The function f is component-wise strictly quasi-convex with respect to the two blocks $\boldsymbol{L}$ and $\boldsymbol{R}$. This is because $f(\boldsymbol{L}, \boldsymbol{R}, \boldsymbol{\Pi})$ w.r.t. $\boldsymbol{L}$ and $f(\boldsymbol{L}, \boldsymbol{R}, \boldsymbol{\Pi})$ w.r.t. $\boldsymbol{R}$ are convex terms it follows that they are pseudo-convex functions [89] and this implies they are also strictly quasi-convex functions [89]. It then follows from [90, Prop. 5] that the sequence generated by alternating minimization $\{(\boldsymbol{L}_t, \boldsymbol{R}_t, \boldsymbol{\Pi}_t)\}_{t=1}^T$ converges to limit points that are also critical points of (3.37). □

3.4 Empirical Results & Discussion

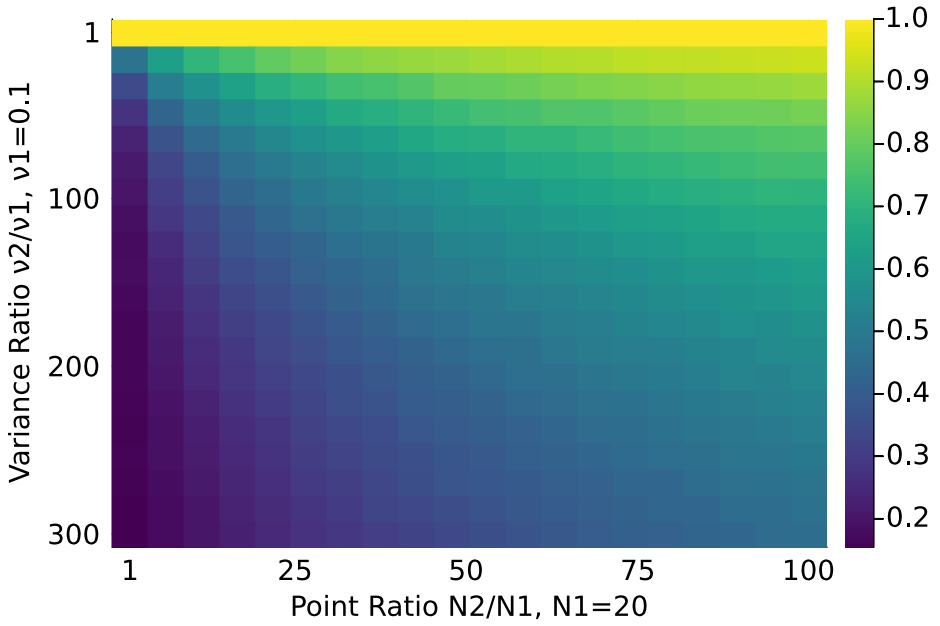
This section summarizes synthetic and real data experiments, including astronomy spectra and RNA sequencing data, that explore various aspects of subspace learning from heteroscedastic data. Code related to these methods and experiments is included at github.com/javiersc1/ALPCAH.

3.4.1 Synthetic Experiments

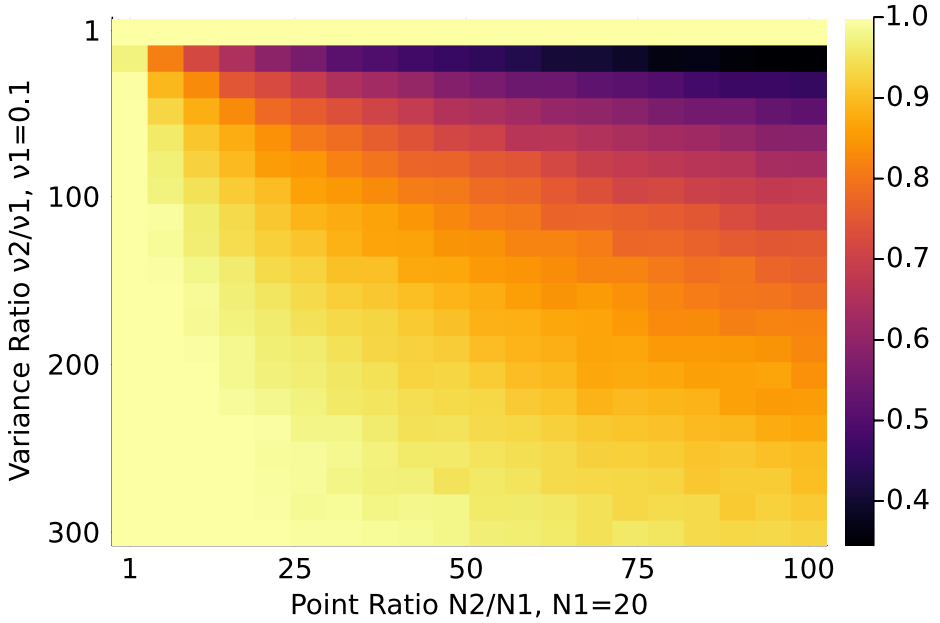
This section uses synthetic data to compare LR-ALPCAH with other methods. We begin by describing the experimental setup, followed by an investigation of PCA, and after that, compare to RPCA, HePPCAT, and WPCA.

Experimental Setup We consider two groups of data, one with fixed quality, meaning fixed size and additive noise variance, and one whose parameters we vary. Let $\boldsymbol{y}_i \in \mathbb{R}^{100}$ be $D = 100$ dimensional ambient-space data. Let $\boldsymbol{U} \in \mathbb{R}^{100\times 5}$ denote a basis for a $d = 5$ dimensional subspace generated by random uniform matrices such that $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}' = \text{svd}(\boldsymbol{A})$, where $A_{ij} \sim \mathcal{U}[0, 1]$. We use the compact SVD here. The low-rank data is simulated as $\boldsymbol{x}_i = \boldsymbol{U}\boldsymbol{z}_i$ where the coordinates $\boldsymbol{z}_i \in \mathbb{R}^5$ were generated from $\mathcal{U}[-100, 100]$ for each element. Then, we generated $\boldsymbol{y}_i = \boldsymbol{U}\boldsymbol{z}_i + \boldsymbol{\epsilon}_i$ where $\boldsymbol{\epsilon}_i \in \mathbb{R}^{100}$ was drawn from $\mathcal{N}(\boldsymbol{0}, \nu_i \boldsymbol{I})$. The error metric used is subspace affinity error (SAE) that compares the difference in projection matrices

$$\text{SAE}(\boldsymbol{U}, \hat{\boldsymbol{U}}) = \|\boldsymbol{U}\boldsymbol{U}' - \hat{\boldsymbol{U}}\hat{\boldsymbol{U}}'\|_{\text{F}} / \|\boldsymbol{U}\boldsymbol{U}'\|_{\text{F}} \tag{3.38}$$



(a) Ratio of subspace affinity errors LR-ALPCAH/PCA (known variance, no cross-validation required)



(b) Ratio of subspace affinity errors LR-ALPCAH/PCA-GOOD (PCA using good data only and LR-ALPCAH using all of the data)

Figure 3.2: Subspace affinity error $\|\boldsymbol{U}\boldsymbol{U}' - \hat{\boldsymbol{U}}\hat{\boldsymbol{U}}'\|_{\mathrm{F}} / \|\boldsymbol{U}\boldsymbol{U}'\|_{\mathrm{F}}$ performance of LR-ALPCAH compared to PCA.

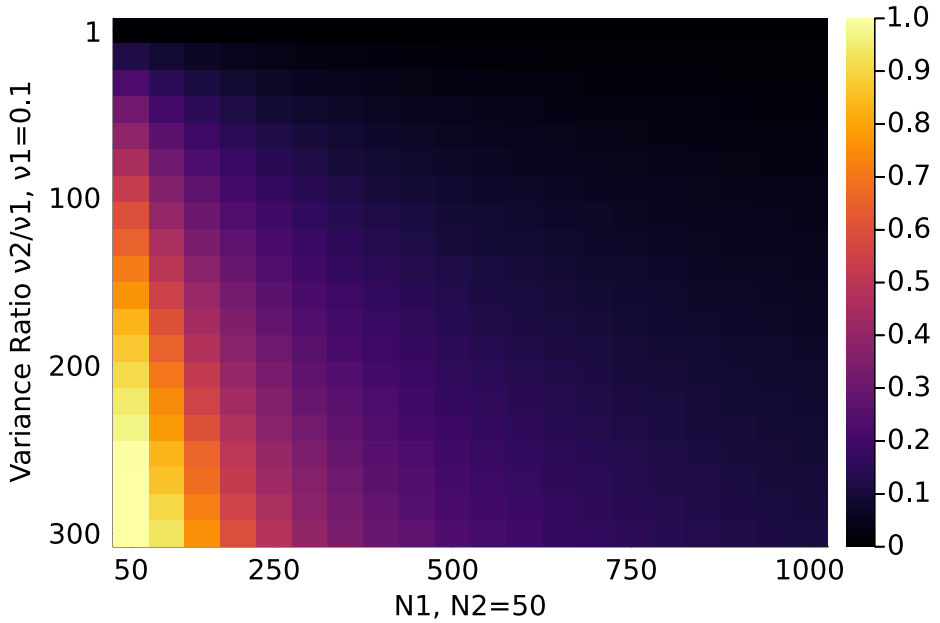


Figure 3.3: Absolute difference of LR-ALPCAH subspace error subtracted from PCA while the amount of good data varies.

so that a low error signifies a closer estimate of the true subspace. This metric is also known as normalized chordal distance [91]. In summary, the noisy data $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N]$ is generated accordingly, an estimate $\hat{\boldsymbol{X}}$ is generated from (3.1), the subspace basis is calculated by $\hat{\boldsymbol{X}} = \sum_i \hat{\sigma}_i \hat{\boldsymbol{u}}_i \hat{\boldsymbol{v}}_i{}' \implies \hat{\boldsymbol{U}} = [\hat{\boldsymbol{u}}_1, \ldots, \hat{\boldsymbol{u}}_d]$, and we report the subspace affinity error.

Subspace Basis Estimation (LR-ALPCAH vs. PCA) We explored the effects of data quality and data quantity on the heteroscedastic subspace basis estimates in different situations. For the heatmaps in Fig. 3.2, we focused on comparing LR-ALPCAH with PCA only to discuss this method in the general context of subspace learning. In Fig. 3.2, each pixel represents the ratio $\text{SAE}(\boldsymbol{U}, \hat{\boldsymbol{U}}_{\text{LR-ALPCAH}})/\text{SAE}(\boldsymbol{U}, \hat{\boldsymbol{U}}_{\text{PCA}})$. A value close to 1 implies LR-ALPCAH did not perform much better than the other method, whereas a ratio closer to 0 implies LR-ALPCAH performed relatively well. The average SAE ratio of 50 trials is used, where each trial has different noise, basis coefficients, and subspace basis realizations. The noise variance for group 1 is fixed to $\nu_1 = 0.1$ with $N_1 = 20$ point samples. We varied group 2 point samples N_2 and noise variances ν_2, as illustrated in the x-axis and y-axis, respectively, for the heatmaps shown.

Fig. 3.2a compares LR-ALPCAH against PCA in the situation where noise variances are known. In this case, LR-ALPCAH performs well relative to PCA in noisy situations and can improve estimation, especially in extreme heteroscedastic regions. From the bottom left corner and moving rightwards, the estimation error worsened as the number of noisy points increased. To clarify, LR-ALPCAH never performed worse than PCA, only that the advantage gap decreased

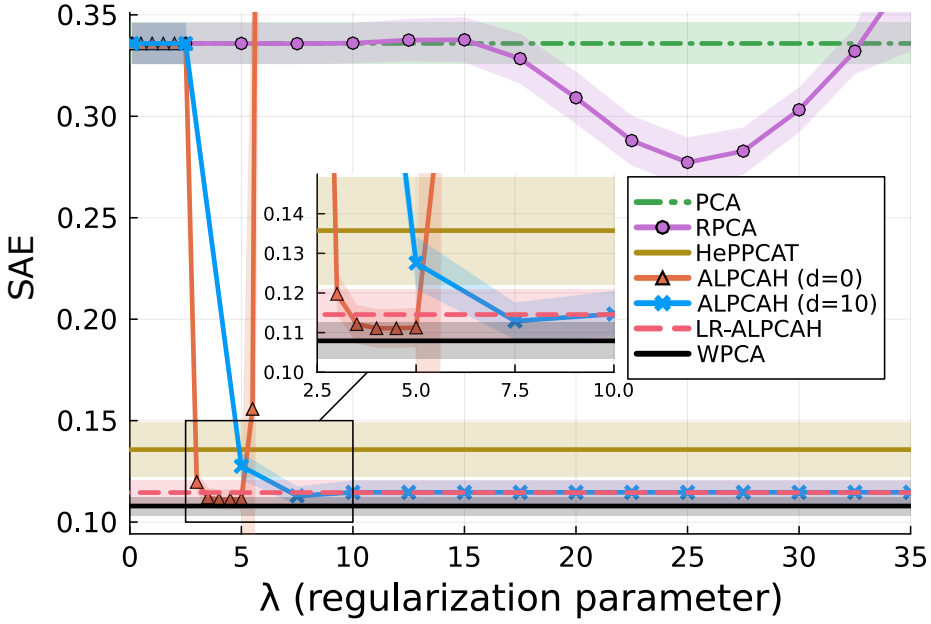


Figure 3.4: Absolute subspace quality performance of ALPCAH compared against other methods. Zoomed-in areas shown within plots for better visibility for certain λ ranges.

as more noisy samples were added. This means that, in LR-ALPCAH, the noisy points may have contributed too much to the estimation process, while good-quality data should have had more influence. For these results, we used the inverse noise variances as the weighing scheme as this is a natural choice that arises from the Gaussian likelihood. However, finding the optimal scheme to mitigate this worsening effect is a topic of future work.

Fig. 3.2b is similar to Fig. 3.2a but only using the high-quality points for PCA specifically, whereas LR-ALPCAH used all of the data. One can see that even when there was enough good data, there was still an improvement over applying PCA to just the good data. The improvement increased as more noisy points were added. Thus, it is beneficial to collect and use all of the data, since the noisy points offer meaningful information that can improve the estimate of the basis versus using good data alone, especially in data-constrained situations.

Effects of Good Data Fig. 3.3 explores how the number of good data samples affects subspace learning quality. We fixed $N_2 = 50$ and varied N_1 while keeping $\nu_1 = 0.1$ and varying ν_2. This figure plots the difference $\text{SAE}(\boldsymbol{U}, \hat{\boldsymbol{U}}_{\text{PCA}}) - \text{SAE}(\boldsymbol{U}, \hat{\boldsymbol{U}}_{\text{LR-ALPCAH}})$ to see when it is advantageous to use LR-ALPCAH instead of PCA. In the absolute sense, both methods performed similarly when good data is abundant. However, when good data was more limited, there were larger differences in subspace quality, meaning it is more advantageous to use LR-ALPCAH.

Absolute Subspace Error This section discusses the absolute errors of the algorithms in the unknown noise variance setting without group knowledge. For Fig. 3.4, we fixed $N_1 = 50, N_2 = 450$ that have noise variances $\nu_1 = 0.25, \nu_2 = 100$. The regularization parameter λ is varied (ALPCAH & RPCA only), and the subspace dimension is $d = 10$. As before, we use the subspace affinity error $\text{SAE}(\boldsymbol{U}, \hat{\boldsymbol{U}})$. The average error is plotted out of 50 trials with standard deviation bounds for each λ value. Fig. 3.4 represents the unknown variance case, but we again use WPCA with weights $w_i = \nu_i^{-1}$, a known variance method, to illustrate the lowest possible affinity error if one hypothetically knew the noise variances.

In Fig. 3.4, when using rank knowledge, ALPCAH ($\hat{d} = 10$) approaches the error of the other methods as λ grows. When not using rank knowledge, for ALPCAH ($\hat{d} = 0$), the method can perform just as well but requires cross-validation to find an ideal λ range. Both ALPCAH ($\hat{d} = 10$) and LR-ALPCAH achieved lower error than HePPCAT, likely because there are no distributional assumptions on the basis coefficients with ALPCAH/LR-ALPCAH. The RPCA method did not perform well in these experiments, likely because of a mismatch between the heteroscedastic data and the RPCA's outlier assumption. Excluding the case when rank knowledge is not known, ALPCAH ($\hat{d} = 0$), the regularization parameter appears to be robust to this landscape of different variance and point ratios.

3.4.2 Real Data Experiments

3.4.2.1 Astronomy spectra data

We investigated quasar spectra data from the Sloan Digital Sky Survey (SDSS) Data Release 16 [92] using its DR16Q quasar catalog [93]. Each quasar has a vector of flux measurements across wavelengths that describes the intensity of observing that particular wavelength. In this dataset, the noise is heteroscedastic across the sample space (quasars) and feature space (wavelength), but we focused on a subset of data that is homoscedastic across wavelengths and heteroscedastic across quasars. The noise for each quasar is known given the measurement devices used for data collection [92], but we performed estimation as if the variances were unknown so that we could compare the estimated values to the reference values. We preprocessed the data (filtering, interpolation, centering, and normalization) based on supplementary material 5 of [94]. We formed a training dataset based on the 1000 smallest variance quasar flux samples and performed PCA to get a "ground-truth" measurement of the subspace basis using $\hat{d} = 5$ as the rank parameter estimated from SignFlipPA [79]. We formed the test dataset by excluding the 1000 samples used during training and combining 9000 samples of various noise quality, leading to heteroscedasticity across samples as shown in Fig. 3.5. This figure shows only the 3000 lowest noise variance data samples along with 2000 noisier samples to illustrate the differences in data quality.

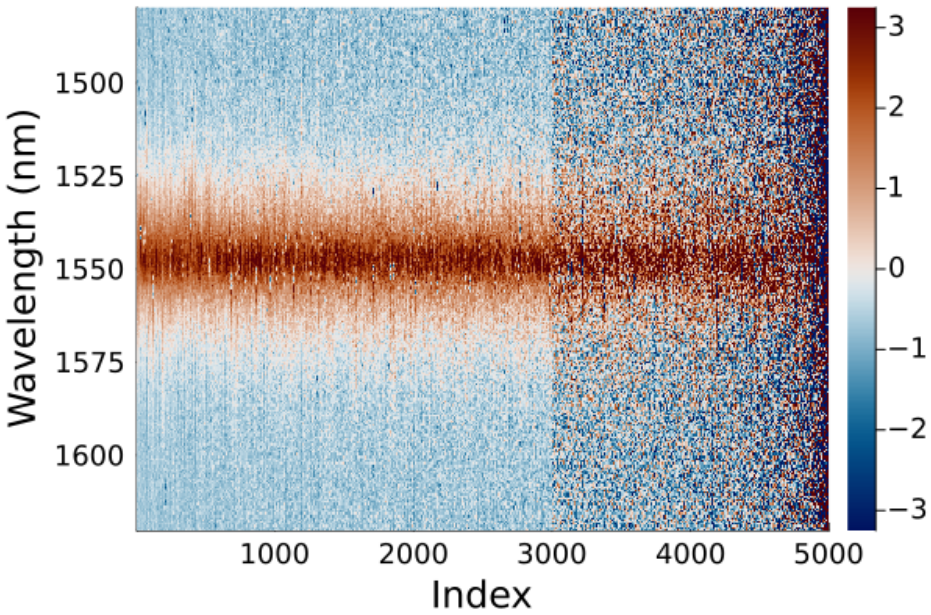
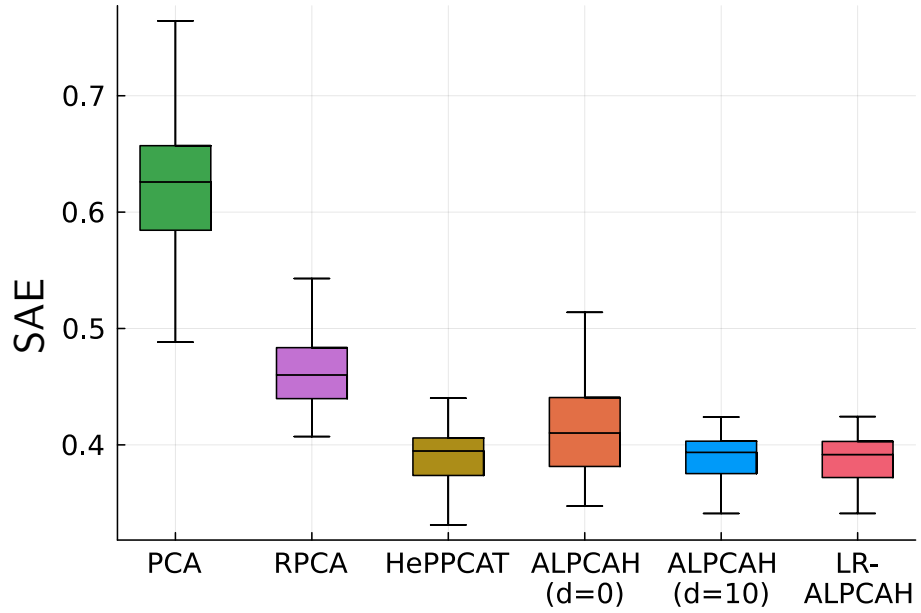


Figure 3.5: Sample data matrix of quasar flux measurements across wavelengths for each (column-wise) sample.

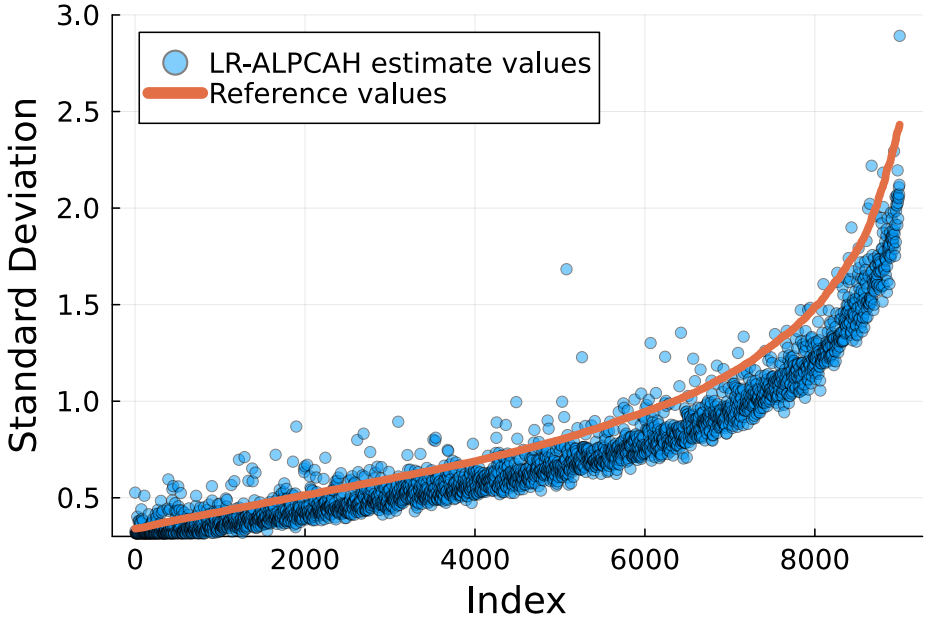
For our subspace quality experiment, we report SAE using the "ground-truth" basis over 100 trials for various methods. We ran each applicable subspace learning algorithm for 100 iterations to ensure convergence. In Fig. 3.6a, RPCA seems to perform slightly worse than the other methods, indicating a model mismatch between outliers and heteroscedastic data. Moreover, it seems that LR-ALPCAH and ALPCAH performed equally well as HePPCAT in this specific real data example. All methods performed better than PCA, indicating a mismatch between the homoscedastic assumption of PCA and the heteroscedastic data. Additionally, we examined the computational time and memory requirements for these methods on this test dataset. Table 3.1 shows that the proposed LR-ALPCAH method is both extremely fast and memory efficient relative to the other heteroscedastic methods, as shown in bold. Since we have reference noise variance values, we also examined how the estimated noise variance values compared to the reference values. Fig. 3.6b sorts the data based on the reference variance values and plots the ALPCAH estimates. ALPCAH estimates generally tracked the global trend found in the reference values but there are minor variations among adjacent points.

3.4.2.2 Biological scRNA-seq data

This section applies PCA and LR-ALPCAH to real data from single-cell RNA-sequencing data (scRNA-seq) from [95]. This sequencing technology is useful for quantifying the transcriptome of individual cells [96]. The data is high-dimensional since thousands of genes are counted for



(a) Subspace learning results showing SAE for ALPCAH and LR-ALPCAH relative to other methods.



(b) Noise standard deviations estimates of LR-ALPCAH compared to known astronomical reference values.

Figure 3.6: Experimental results of quasar flux data for subspace learning and noise sample estimation.

	Time (ms)	Memory (MiB)	Mean SAE
Homoscedastic Reference			
PCA	32.4	7.3	0.65
Heteroscedastic Methods			
RPCA (classical)	5091.8	7977.5	0.46
HePPCAT	1339.1	5731.6	0.39
ALPCAH ($\hat{d}$=0)	4339.3	3838.6	0.41
ALPCAH ($\hat{d}$=10)	4339.9	3838.8	0.39
LR-ALPCAH	**153.5**	**459.0**	**0.38**

Table 3.1: Subspace learning results on quasar flux data.

thousands of cell samples, which produces challenges for data analysis. PCA methods are useful for scRNA-seq data to perform gene variation analysis and clustering in low-dimensional spaces to study gene groups [97]. Heterogeneous noise may occur among both cells and genes [79], which prompts further investigation into heteroscedastic-aware PCA methods on scRNA-seq data. The data matrix consists of 10,000 cells by 5,000 genes. We preprocessed the data by subtracting the mean and replacing the missing values with zeros. Since the noise variances are unknown in this application, we cannot have a "ground truth" subspace to compare against. Instead, we separate the data into train and test, and calculate the NRMSD to compare reconstruction quality, i.e.,

$$\text{NRMSD} = \|\boldsymbol{Y}_{\text{test}} - \boldsymbol{U}_{\text{train}}\boldsymbol{U}_{\text{train}}'\boldsymbol{Y}_{\text{test}}\|_{\text{F}} \;/\; \|\boldsymbol{Y}_{\text{test}}\|_{\text{F}}. \tag{3.39}$$

The subspace basis was learned on the training data with PCA or LR-ALPCAH and the test data was used to assess reconstruction quality by projecting test data onto the subspace basis and using the basis coefficients to return to the ambient space. In this experiment, SignFlipPA was used to determine an appropriate rank [79]. Fig. 3.7 shows a subset of the data matrix. Here, the color map is clipped to 10 to better visualize the matrix as most gene counts are sparse. The middle plot shows sorted noise variances estimated by LR-ALPCAH indicating some potential heterogeneity by one or two orders of magnitude. The right plot shows that LR-ALPCAH has a better reconstruction quality since it has $\sim$0.1 lower NRMSD than PCA. The difference between PCA and LR-ALPCAH is more modest with this dataset. Possibly, the results could be improved further by developing a method that handles heteroscedasticity across both the samples and features, as this data is doubly heteroscedastic. Moreover, real scRNA-seq data pose additional challenges, such as dependent noise that our method does not model. However, preliminary results indicate that LR-ALPCAH is a promising approach and further investigation into addressing model assumptions is an interesting direction of future work.

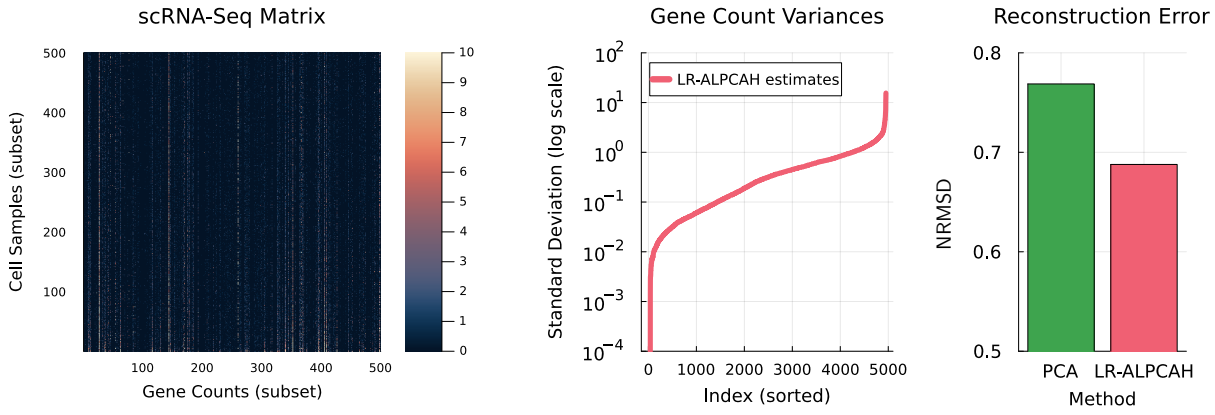


Figure 3.7: Biological scRNA-seq data results.

3.5 Conclusion

This chapter proposed two subspace learning algorithms that are robust to heteroscedasticity by jointly learning the noise variances and subspace bases. While LR-ALPCAH is memory-efficient and fast, its application is limited to sample-wise heteroscedasticity. It would be interesting to generalize this work to be doubly heteroscedastic, where the features themselves also have different noise variances. Applications such as biological sequencing [65] and photon imaging [98] could benefit from such an extension. In the scRNA-seq application, we computed missing entries as zeros, which is a natural choice for low-rank models. However, others have worked on adapting PCA methods for missing data [99], so such an approach could be beneficial given the higher than expected NRMSD with LR-ALPCAH. This generalization is nontrivial so it is left for future work. Additionally, our model and the comparison methods are limited to the subspace setting, but some applications like resting-state functional MRI [100] benefit from manifold learning approaches [100]. It would be interesting to explore other approaches such as a heteroscedastic variational autoencoder [101] to expand the range of applications for heteroscedastic data learning.

3.6 Additional Results

3.6.1 PCA Bound Experiment

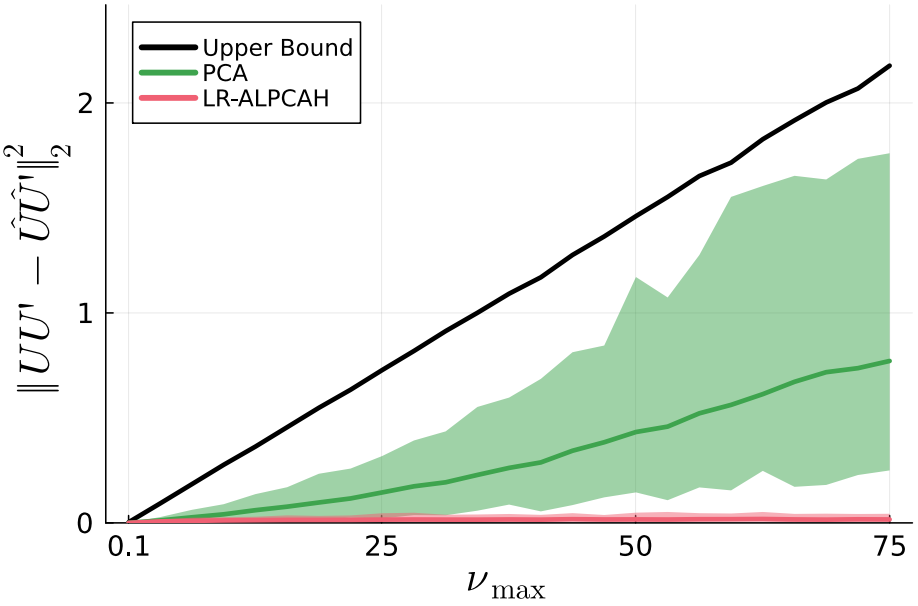


Figure 3.8: Experimental verification of heteroscedastic impact on PCA upper bound (3.7).

This section focuses on providing empirical verification of our subspace bound in (3.7). Before doing so, we mention that the random matrix theory bound in (3.3) depends on a universal constant c_1, independent of D and N, that is not calculated in the source paper [72]. Let $A = [a]$ be a 1×1 matrix with element $a \sim \mathcal{N}(0, 1)$. For the LHS in (3.3), the spectral norm in this instance is $\|A\|_2 = |a|$. This implies that $\mathbb{E}[\|A\|_2]$ is equivalent to calculating the mean of a folded normal distribution. Since a is a standard normal random variable, $\mathbb{E}[|a|] = \sqrt{2/\pi}$. One can verify from (3.4) (3.5) (3.6) that the RHS in (3.3) simplifies to $2 + \sqrt[4]{3}$. Solving for c_1, the inequality becomes $c_1 \geq \sqrt{2/\pi}/(2 + \sqrt[4]{3}) \approx 0.24$. In our subspace bound (3.7), both sides are squared and a factor of 2 exists in (3.2), therefore the constant in (3.7) is $c = 4c_1^2 \approx 0.22$. Knowing this constant, it is now possible to experimentally verify (3.7). The experimental setup consists of generating random rank-3 subspaces within a 100 dimensional ambient space. The data samples consist of two groups, one with $n_1 = 30$ samples, $\nu_1 = 0.1$ and the other with $n_2 = 970$ samples and a varying $\nu_2 \in \{0.1, \ldots, 75\}$. During the course of 50 trials, we computed the mean spectral norm projection error, i.e., $\|\hat{\boldsymbol{U}}\hat{\boldsymbol{U}}' - \boldsymbol{U}\boldsymbol{U}'\|_2^2$, along with the minimum and maximum error values for that ν_2 instance. Fig. 3.8 illustrates that PCA scales similarly to the bound in (3.7), yet our method, LR-ALPCAH, empirically did not degrade at the same rate, indicating robustness to heteroscedasticity.

3.6.2 Matrix Factorized Robust PCA

In Sec. 3.2.4, RPCA is introduced with a convex formulation that involves the nuclear norm. This norm, while convex, is a soft constraint on the rank of the low-rank matrix $\boldsymbol{X}$. However, since LR-ALPCAH performs well in the matrix factorized setting, it is also interesting to explore a matrix factorized RPCA formulation to augment the results in Fig. 3.4. Therefore, instead of assuming that $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{E}$, one can instead assume that $\boldsymbol{Y} = \boldsymbol{L}\boldsymbol{R}' + \boldsymbol{E}$ where $\boldsymbol{L}, \boldsymbol{R}$ contain $\hat{d}$ columns and generally $\hat{d} \ll \min(D, N)$. With this in mind, we formulate a matrix factorized version of RPCA as follows

$$\underset{\boldsymbol{L},\boldsymbol{R},\boldsymbol{E}}{\arg\min} \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{L}\boldsymbol{R}' - \boldsymbol{E}\|_{\mathrm{F}}^2 + \lambda\|\boldsymbol{E}\|_{1,1}. \tag{3.40}$$

Since the cost function is now non-convex, it is worth initializing in a smart way as opposed to randomly. Just like with LR-ALPCAH, one can use the spectral init approach in (3.33) to compute $\boldsymbol{L}_0$ and $\boldsymbol{R}_0$, and then compute $\boldsymbol{E}_0 = \boldsymbol{Y} - \boldsymbol{L}_0\boldsymbol{R}_0'$ to initialize $\boldsymbol{E}$. From there, we used alternating minimization to generate update formulas for all of the three blocks.

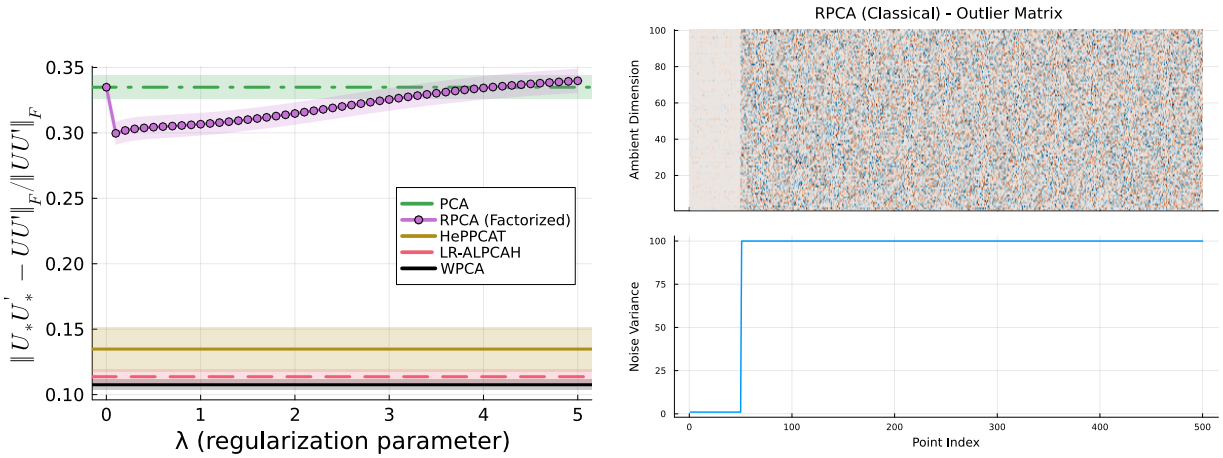
With this formulation in mind, the same methodology can be applied from Fig. 3.4 to generate a similar figure for this matrix factorized version of RPCA. In Fig. 3.9, one can see that the matrix factorized RPCA method performs worse than the classical RPCA method introduced in Sec. 3.2.4, likely because of the non-convex landscape and the model mismatch between sparse outliers and heteroscedasticity. This performance difference remains even when the appropriate rank is provided to the method as illustrated in Fig. 3.9. Additionally, the outlier matrix does appear to mainly capture information from the noisy samples, with some outliers captured in the low noise group.

3.6.3 Group Sparsity Robust PCA

In Sec. 3.2.4, RPCA is introduced with an $\ell_{1,1}$ norm to promote sparsity and capture outliers inherent in the data. This is the traditional formulation used in the original paper [69]. However, one can consider "group sparsity" or $\ell_{2,1}$ norm minimization in this case to promote sparsity only on samples that are excessively noisy and not on the low noise group. Therefore, we formulate a group sparsity RPCA as follows

$$\underset{\boldsymbol{X},\boldsymbol{E}}{\arg\min} \left(\lambda\|\boldsymbol{X}\|_* + \|\boldsymbol{E}\|_{2,1}\right) \text{ s.t. } \boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{E}. \tag{3.41}$$

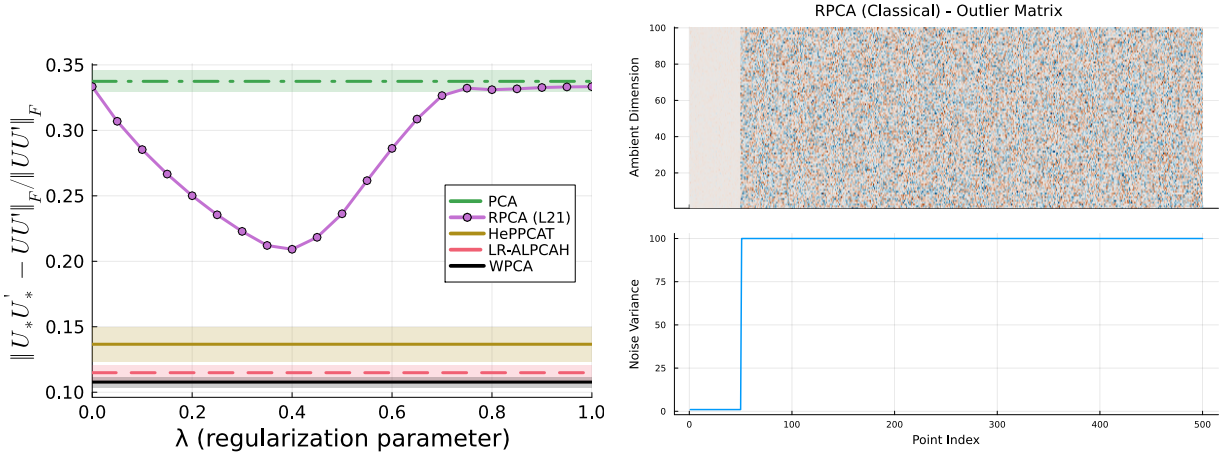
Sec. 3.6.2 and Sec. 3.6.3 are new extensions to the TSP paper [13].



(a) Absolute subspace quality performance.

(b) Outlier matrix estimate $\hat{\boldsymbol{E}}$ when $\lambda = 0.01$.

Figure 3.9: Matrix factorized RPCA results.



(a) Absolute subspace quality performance.

(b) Outlier matrix $\hat{\boldsymbol{E}}$ when $\lambda = 0.4$.

Figure 3.10: Group Sparsity RPCA results.

Just like with L11 norm minimization, L21 norm minimization also entails a proximal mapping whose solution is closed-form, and involves soft thresholding specific columns of a matrix, as opposed to the entire matrix. With this formulation in mind, the same methodology can be applied from Fig. 3.4 to generate a similar figure for this group sparsity version of RPCA. Fig. 3.10 illustrates that the group sparsity version performs better than classical RPCA, since the mean subspace affinity error is 0.22 instead of 0.28. By analyzing the outlier matrix, it is clear that no outliers are captured in the low noise group, as we would expect. However, the results pale in comparison to heteroscedastic-focused methods like LR-ALPCAH and HePPCAT. Thus, there is still a general model mismatch between outlier modeling and heteroscedastic modeling, leading to slightly worse results for RPCA, even with group sparsity considered.

CHAPTER 4

ALPCAHUS: Subspace Clustering for Heteroscedastic Data

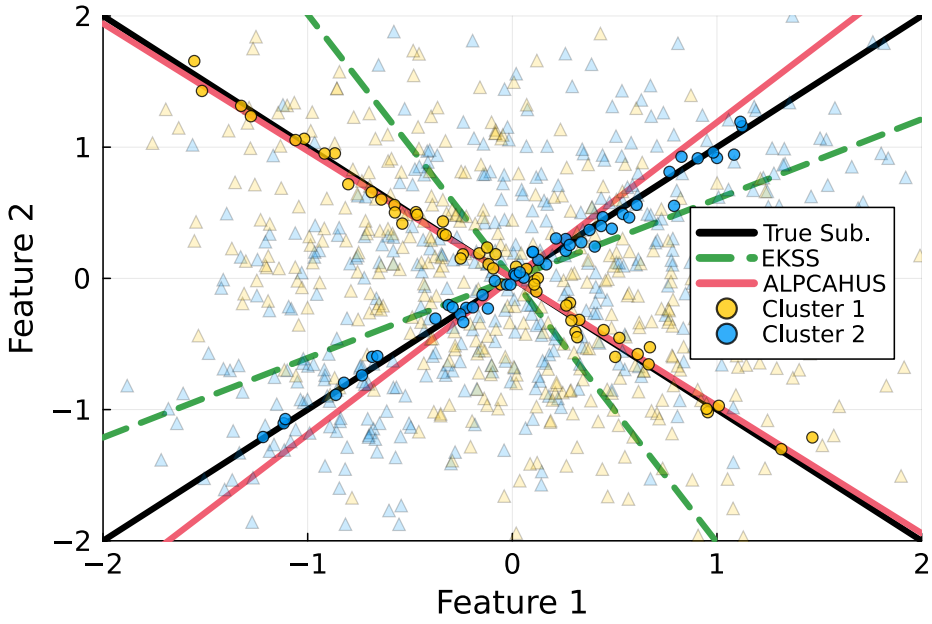


Figure 4.1: Two 1D subspaces, colored blue and yellow, with data consisting of two noise groups shown with circle (low noise) and triangle (high noise) markers.

4.1 Introduction

Many modern data science problems require learning an approximate signal subspace basis for some collection of data. This is important for downstream tasks involving subspace basis coefficients such as classification [56], regression [57], and compression [58]. Besides subspace

The work in this chapter is under review by the IEEE Transactions on Signal Processing (TSP) [14].

learning, one may be interested in clustering data points that originate from multiple subspaces. Formally, subspace clustering, or union of subspace (UoS) modeling, is an unsupervised machine learning problem where the goal is to cluster unlabeled data and find the subspaces associated with each data cluster. When the cluster assignments are known, it is easy to find the subspaces, and vice versa. This problem becomes nontrivial when both components must be estimated [102]. This clustering problem has many applications, such as image segmentation [103], motion segmentation [104], image compression [105], and system identification [106].

Some applications involve heterogeneous data samples that vary in quality due in part to noise characteristics associated with each sample. A few examples of heteroscedastic datasets include environmental air quality data [63], astronomical spectral data [64], and biological sequencing data [65]. In heteroscedastic settings, the noisier data samples can significantly corrupt the basis estimates [66]. In turn, this corruption can worsen clustering performance as seen in Fig. 4.1. Popular clustering methods such as Sparse Subspace Clustering (SSC) [23], K-Subspaces (KSS) [24], and Subspace Clustering via Thresholding (TSC) [107], all implicitly assume that data quality is consistent. For example, in SSC, the method relies on the self-expressiveness property of data that uses other similar samples to estimate every single sample. From our experiments, we found that this implicit data quality assumption can degrade clustering quality for heteroscedastic data.

Because of these limitations, we developed a subspace clustering algorithm, inspired by the K-Subspaces method, that explicitly models noise variance terms, without assuming that data quality is known. The method adaptively clusters data while learning noise characteristics. See Fig. 4.1 for a visualization where Ensemble KSS (EKSS) [108] returns poor subspace bases estimates whereas our method found more accurate subspace bases and improved clustering quality.

We extend our previous work [13] by generalizing the LR-ALPCAH formulation to the UoS setting for clustering heteroscedastic data. The proposed approach achieved $\sim$3 times lower clustering error than existing methods, and it achieved a relatively low clustering error even when very few high-quality samples were available. This section is divided into a few key subsections. Sec. 4.2 introduces the heteroscedastic problem formulation for subspace clustering. Sec. 4.3 discusses related work in subspace clustering and reviews the heteroscedastic subspace algorithm LR-ALPCAH. Sec. 4.4 introduces the proposed subspace clustering method named ALPCAHUS. Sec. 4.5 covers synthetic and real data experiments that illustrate the effectiveness of modeling heteroscedasticity in a clustering context. Finally, Sec. 4.6 discusses some limitations of our method and possible extensions.

4.2 Problem Formulation

Let K denote the number of subspaces that are either known beforehand or estimated using other methods. Before describing the general union-of-subspaces model, the single-subspace model under $K = 1$ is introduced.

4.2.1 Single Subspace Model ($K = 1$)

Let $\boldsymbol{y}_i \in \mathbb{R}^D$ denote the data samples for index $i \in \{1, \ldots, N\}$, where D denotes the ambient dimension and N is the total number of samples. Let $\boldsymbol{x}_i$ represent the low-dimensional data sample generated by $\boldsymbol{x}_i = \boldsymbol{U}\boldsymbol{z}_i$ where $\boldsymbol{U} \in \mathbb{R}^{D\times d}$ is an unknown basis for a subspace of dimension d and $\boldsymbol{z}_i \in \mathbb{R}^d$ are the corresponding basis coordinates. Then, the heteroscedastic model we consider is

$$\boldsymbol{y}_i = \boldsymbol{x}_i + \boldsymbol{\epsilon}_i \quad \text{where} \quad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\boldsymbol{0}, \nu_i \boldsymbol{I}) \tag{4.1}$$

assuming Gaussian noise with variance ν_i, where $\boldsymbol{I}$ denotes the $D \times D$ identity matrix. In this work, we consider the case where each data sample may have its own noise variance. However, one can adapt this method to consider the case where there are G groups of data having shared noise variance terms $\{\nu_1, \ldots, \nu_G\}$. Sec. 4.3.3 discusses an optimization problem based on this model that estimates the heterogeneous noise variances $\{\nu_i\}_{i=1}^N$ and the subspace basis $\boldsymbol{U}$.

4.2.2 Union of Subspaces Model ($K \geq 1$)

Let $\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}_1 & \ldots & \boldsymbol{y}_N \end{bmatrix} \in \mathbb{R}^{D\times N}$ denote a matrix whose columns consist of all N data points $\boldsymbol{y}_i \in \mathbb{R}^D$. We generalize (4.1) to model the data with a union of subspaces model by

$$\begin{aligned} \boldsymbol{y}_i &= \boldsymbol{x}_i + \boldsymbol{\epsilon}_i \\ \boldsymbol{x}_i &= \boldsymbol{U}_{k_i}\boldsymbol{z}_i \text{ for some } k_i \in \{1, \ldots, K\}, \end{aligned} \tag{4.2}$$

where $\boldsymbol{U}_k \in \mathbb{R}^{D\times d_k}$ is a subspace basis that has subspace dimension d_k. Here, $\boldsymbol{z}_i \in \mathbb{R}^{d_k}$ denotes the basis coefficients associated with $\boldsymbol{x}_i$, and $\boldsymbol{\epsilon}_i \in \mathbb{R}^D$ denotes noise for that point drawn from $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\boldsymbol{0}, \nu_i \boldsymbol{I})$.

If the subspace bases were known, then one would like to find the associated subspace label $c_i \in \{1, \ldots, K\}$ for each data sample by solving the following optimization problem

$$c_i = \arg\min_k \|\boldsymbol{y}_i - \boldsymbol{U}_k \boldsymbol{U}_k' \boldsymbol{y}_i\|_2^2, \quad \forall \boldsymbol{y}_i \in \boldsymbol{Y}. \tag{4.3}$$

This label describes the subspace association of a data point $\boldsymbol{y}_i$ that has the lowest residual between the original sample and its reconstructed value given the subspace. In general, the goal is to estimate all of the subspace bases $\mathcal{U} = (\boldsymbol{U}_1, \ldots, \boldsymbol{U}_K)$ and cluster assignments $\mathcal{C} = (c_1, \ldots, c_N)$ associated with all the data samples.

4.3 Related Works

4.3.1 Subspace Clustering

Many subspace clustering algorithms fall into a general umbrella of categories such as algebraic methods [109], iterative methods [110], statistical methods [111], and spectral clustering methods [112] [113]. In recent years, both spectral clustering-based methods and iterative methods have become popular.

4.3.1.1 Spectral Clustering

Many methods build on spectral clustering. This method is often used for clustering nodes in graphs. Spectral clustering aims to find the minimum cost "cuts" in the graph to partition nodes into clusters. Given some collection of data points, one way to construct a graph is to assume that "nearby" data samples are highly connected in the graph. Thus, it is possible to construct a graph from data samples with varying levels of connectedness by using a metric to assign affinity/similarity to each pair of points. Let $\mathcal{G}(\mathcal{V}, \boldsymbol{W})$ correspond to a graph that consists of vertices $\mathcal{V} = \{v_1, \ldots, v_N\}$ and edge weights $\boldsymbol{W} \in \mathbb{R}^{N \times N}$ such that w_{ij} corresponds to some nonnegative weight, or similarity, between v_i and v_j. The graph $\mathcal{G}$ is assumed to be undirected, i.e., $\boldsymbol{W} = \boldsymbol{W}'$. The degree of a node is defined as $d_i = \sum_{j=1}^{N} w_{ij}$ and can be collected to form a degree matrix such that $\boldsymbol{D} = \text{diag}(d_1, \ldots, d_N)$. Let $\mathcal{A}_1, \ldots, \mathcal{A}_K$ form a K-partition on $\mathcal{G}(\mathcal{V}, \boldsymbol{W})$ and $\text{cut}(\mathcal{A}, \mathcal{B}) = \sum_{i \in \mathcal{A}, j \in \mathcal{B}} w_{ij}$. Then, spectral clustering aims to solve the following

$$\underset{\mathcal{A}_1, \ldots, \mathcal{A}_K}{\arg\min} \frac{1}{2} \sum_{i=1}^{K} \frac{\text{cut}(\mathcal{A}_i, \bar{\mathcal{A}}_i)}{|\mathcal{A}_i|} \tag{4.4}$$

where $\bar{\mathcal{A}}_i$ is all vertices not belonging in $\mathcal{A}_i$ and $|\mathcal{A}_i|$ is the number of vertices belonging to the ith partition. Instead of working with $\boldsymbol{W}$ directly, one computes a normalized graph Laplacian such as $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-1}\boldsymbol{W}$ to make the influence of heavy degree nodes more similar to low degree nodes.

Let $h_{i,j}$ indicate the jth cluster membership for the ith point by

$$h_{ij} = \begin{cases} 1/\sqrt{|\mathcal{A}_j|} & \text{if } v_i \in \mathcal{A}_j \\ 0 & \text{otherwise.} \end{cases} \tag{4.5}$$

Collecting the indicator values into a matrix $\boldsymbol{H} \in \mathbb{R}^{N \times K}$, one can rewrite (4.4) in matrix notation as

$$\frac{1}{2} \sum_{i=1}^{K} \frac{\text{cut}(\mathcal{A}_i, \bar{\mathcal{A}}_i)}{|\mathcal{A}_i|} = \sum_{i=1}^{K} (\boldsymbol{H}'\boldsymbol{L}\boldsymbol{H})_{ii} = \text{Tr}(\boldsymbol{H}'\boldsymbol{L}\boldsymbol{H}), \tag{4.6}$$

where $\text{Tr}(\boldsymbol{H}) = \sum_i \boldsymbol{H}_{ii}$ denotes the trace of a matrix. The indicator vectors are discrete, which makes the problem computationally challenging. One way to relax the problem is to allow arbitrary real values for $\boldsymbol{H}$ and instead solve

$$\hat{\boldsymbol{H}} = \underset{\boldsymbol{H} \in \mathbb{R}^{N \times K}}{\arg\min} \text{Tr}(\boldsymbol{H}'\boldsymbol{L}\boldsymbol{H}) \text{ s.t. } \boldsymbol{H}'\boldsymbol{H} = \boldsymbol{I} \tag{4.7}$$

where the notation s.t. means subject to some conditions. Trace minimization problems with semi-unitary constraints are well-studied in the literature, and it is easy to see that $\hat{\boldsymbol{H}}$ consists of the first K eigenvectors of $\boldsymbol{L}$ associated with the smallest eigenvalues, i.e., $\lambda_1(\boldsymbol{L}) \leq \ldots \leq \lambda_N(\boldsymbol{L})$ where $\lambda_i(\boldsymbol{L})$ denotes the ith eigenvalue of $\boldsymbol{L}$. Once $\hat{\boldsymbol{H}}$ is computed, the K-means algorithm is applied to $\hat{\boldsymbol{H}}'$, treating each row of $\hat{\boldsymbol{H}}$ as the spectral embedding of the associated v_i vertex. The next section, Sec. 4.3.1.2, now describes some subspace clustering methods that use this technique by first forming a weight matrix $\boldsymbol{W}$ and then applying spectral clustering to $\boldsymbol{W}$.

4.3.1.2 Self-Expressive Methods

Self-expressive methods exploit the "self-expressiveness property" [114] of data that hypothesizes that a single data point can be expressed as a linear combination of other data points in its cluster, which is trivially true if the data exactly follows a subspace model. The goal is to learn those linear coefficients, which is often achieved by adopting different regularizers in the formulation. These self-expressive algorithms, e.g., SSC [23] and LRSC [115], learn the coefficient matrix $\boldsymbol{P} \in \mathbb{R}^{N \times N}$ by solving special cases of the following general optimization problem

$$\underset{\boldsymbol{P}}{\arg\min} \text{DF}(\boldsymbol{Y} - \boldsymbol{Y}\boldsymbol{P}) + \lambda \Lambda(\boldsymbol{P}) \text{ s.t. } \boldsymbol{P} \in \mathcal{S}_{\boldsymbol{P}}. \tag{4.8}$$

The function $\text{DF}(\boldsymbol{Y} - \boldsymbol{Y}\boldsymbol{P})$ is a data fidelity term, $\Lambda(\boldsymbol{P})$ is a regularizer, and $\mathcal{S}_{\boldsymbol{P}}$ is some constrained set to encourage $\boldsymbol{P}$ to satisfy certain conditions. In the case of SSC, the optimization

problem is formulated as

$$\arg\min_{\boldsymbol{P}} \|\boldsymbol{Y} - \boldsymbol{Y}\boldsymbol{P}\|_{\mathrm{F}}^2 + \lambda\|\boldsymbol{P}\|_{1,1} \text{ s.t. } \boldsymbol{P}_{ii} = 0 \ \forall i \tag{4.9}$$

where $\|\boldsymbol{P}\|_{1,1} = \|\text{vec}(\boldsymbol{P})\|_1 = \sum_i |p_i|$ is the vectorized 1-norm and $\|\cdot\|_{\mathrm{F}} = \|\text{vec}(\cdot)\|_2$ is the Frobenius norm of a matrix. Observe that (4.9) approximates each data sample as a sparse linear combination of other data points to form $\boldsymbol{P}$. Let $\text{abs}(\boldsymbol{P})$ represent the element-wise absolute value of matrix $\boldsymbol{P}$. Then, spectral clustering is performed on $\boldsymbol{W} = \frac{1}{2}(\text{abs}(\boldsymbol{P}') + \text{abs}(\boldsymbol{P}))$ by applying the K-means method to the spectral embedding of the affinity matrix $\boldsymbol{W}$. Ideally, (4.9) would select the high-quality data to represent the worst samples in $\boldsymbol{P}$ and this information would be retained in $\boldsymbol{W}$. However, this data quality awareness condition is not guaranteed in self-expressive methods to our knowledge. In our experiments, Fig. 4.2f and Fig. 4.7 show that there must be an issue constructing an ideal $\boldsymbol{P}$ due to worse clustering performance in heteroscedastic conditions.

4.3.1.3 Subspace Clustering via Thresholding (TSC)

In the example to follow, assume that there are two distinct 1-dimensional subspaces. Given two data samples $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$, intuitively their dot product $\langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle$ will be high if $c_i = c_j$, meaning they belong to the same subspace. Likewise, if $c_i \neq c_j$, then $\langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle = 0$ if the subspaces are orthogonal. In non-orthogonal scenarios, one expects $|\langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle|$ to be smaller when $c_i \neq c_j$ than when $c_i = c_j$. Using this idea, in more general D-dimensional subspace settings, TSC constructs a matrix $\boldsymbol{Z} \in \mathbb{R}^{N \times N}$ that describes similarity such that

$$\boldsymbol{Z}_{ij} = \exp(-2\arccos(|\langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle|)) \text{ s.t. } \quad \boldsymbol{Z}_{ii} = 0 \ \forall i. \tag{4.10}$$

This matrix is then thresholded to retain only the top q values for each row, i.e.,

$$\boldsymbol{W} = \arg\min_{\boldsymbol{W}} \|\boldsymbol{W} - \boldsymbol{Z}\|_{\mathrm{F}}^2 \text{ s.t. } \|\boldsymbol{W}_{i,:}\|_0 = q \ \forall i \tag{4.11}$$

where $\|\boldsymbol{W}_{i,:}\|_0 = q$ is the l_0 pseudo-norm that counts the number of nonzero entries. In other words, one constructs $\boldsymbol{W}_{i,:}$ using the q nearest neighbors that correspond to the highest magnitude values. Then, spectral clustering is applied to $\boldsymbol{W}$ to find the cluster associations.

4.3.1.4 Ensemble K-Subspaces (EKSS)

Iterative methods are based on the idea of alternating between cluster assignments and subspace basis approximation, with KSS being highly prominent [116]. The KSS algorithm seeks to solve

$$\underset{\mathcal{C},\mathcal{U}}{\arg\min} \sum_{k=1}^{K} \sum_{c_i=k,\forall i} \|\boldsymbol{y}_i - \boldsymbol{U}_k \boldsymbol{U}_k' \boldsymbol{y}_i\|_2^2. \tag{4.12}$$

The goal is to minimize the sum of residual norms by alternating between performing PCA on each cluster to update $\mathcal{U}$ and using the subspace bases to calculate new cluster assignments $\mathcal{C}$ in a similar fashion to the K-means algorithm. The quality of the solution depends highly on the initialization. Recent work provides convergence guarantees and spectral initialization schemes that provably perform better than random initialization [117]. However, this problem (4.12), is known to be NP-hard [118]. Further, it is prone to local minima [119]. To overcome this, consensus clustering [120] is a tool that leverages information from many trials and combines results together. This approach, known as Ensemble KSS (EKSS) [108], creates an affinity matrix whose (i, j)th entry represents the number of times the two points were clustered together in a trial. Then, spectral clustering is performed on the affinity matrix to get the final clustering from the many base clusterings. The use of PCA makes it challenging to learn clusters and subspace bases in the heteroscedastic regime since PCA implicitly assumes the same noise variance across all samples [68]. The proposed ALPCAHUS approach in Sec. 4.4 builds on KSS, and its ensemble version, by generalizing (4.12) to the heteroscedastic regime while simultaneously learning data quality.

4.3.2 Other Heteroscedastic Models

This chapter focuses on heteroscedastic noise across the data samples. There are other *subspace learning* methods in the literature that explore heteroscedasticity in different ways. For example, HeteroPCA considers heteroscedasticity across the feature space [73]. One possible application of that model is for data that consists of sensor information with multiple devices that naturally have different levels of precision and signal-to-noise ratio. Another heterogeneity model considers the noise to be homoscedastic and instead assumes that the signal itself is heteroscedastic [74]. That work considers applications where the power fluctuating signals, i.e., heteroscedastic signals, are embedded in white Gaussian noise. However, to the authors' knowledge, there are no algorithms for *subspace clustering* that consider feature-space heteroscedasticity. We exclude comparisons with these methods since their models are very different and each have their own applications.

4.3.3 Single Subspace Heteroscedastic PCA

Before extending (4.12) to the heteroscedastic setting, we will review how LR-ALPCAH [13] solves for a single subspace ($K = 1$) in the heteroscedastic setting using the model described in (4.1). For the measurement model $\boldsymbol{y}_i \sim \mathcal{N}(\boldsymbol{x}_i, \nu_i \boldsymbol{I})$ in (4.1), the probability density function for a single data sample $\boldsymbol{y}_i$ following a Gaussian distribution is easily expressed as

$$\frac{1}{\sqrt{(2\pi)^D |\nu_i \boldsymbol{I}|}} \exp\left[-\frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{x}_i)'(\nu_i \boldsymbol{I})^{-1}(\boldsymbol{y}_i - \boldsymbol{x}_i)\right]. \tag{4.13}$$

For independent samples, after dropping constants, the joint log likelihood of all data $\{\boldsymbol{y}_i\}_{i=1}^N$ is the following

$$\sum_{i=1}^{N} -\frac{1}{2}\log|\nu_i \boldsymbol{I}| - \frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{x}_i)'(\nu_i \boldsymbol{I})^{-1}(\boldsymbol{y}_i - \boldsymbol{x}_i). \tag{4.14}$$

Let $\boldsymbol{\Pi} = \mathrm{diag}(\nu_1, \ldots, \nu_N) \in \mathbb{R}^{N \times N}$ be a diagonal matrix representing the typically unknown noise variances. Then, the negative log likelihood in matrix form is

$$\frac{D}{2}\log|\boldsymbol{\Pi}| + \frac{1}{2}\mathrm{Tr}[(\boldsymbol{Y} - \boldsymbol{X})\boldsymbol{\Pi}^{-1}(\boldsymbol{Y} - \boldsymbol{X})']. \tag{4.15}$$

After further manipulation by trace lemmas, we rewrite the negative log-likelihood as

$$\frac{1}{2}\|(\boldsymbol{Y} - \boldsymbol{X})\boldsymbol{\Pi}^{-1/2}\|_{\mathrm{F}}^2 + \frac{D}{2}\log|\boldsymbol{\Pi}|. \tag{4.16}$$

Our previous work used a functional operator similar to the nuclear norm to regularize $\boldsymbol{X}$ and encourage a low-rank solution by penalizing the tail sum of singular values [121] [12]. However, this method, named ALPCAH, was relatively slow due to SVD operations every iteration. To reduce computation and enforce a low-rank solution, the LR-ALPCAH variant [13] took inspiration from the matrix factorization literature [70] and factorized $\boldsymbol{X} \in \mathbb{R}^{D \times N} \approx \boldsymbol{L}\boldsymbol{R}'$ where $\boldsymbol{L} \in \mathbb{R}^{D \times \hat{d}}$ and $\boldsymbol{R} \in \mathbb{R}^{N \times \hat{d}}$ for some rank estimate $\hat{d}$. Using this idea, LR-ALPCAH estimates $\boldsymbol{X}$ by solving for $\boldsymbol{L}$ and $\boldsymbol{R}$, jointly with noise variance matrix $\boldsymbol{\Pi}$, by the following

$$\underset{\boldsymbol{L},\boldsymbol{R},\boldsymbol{\Pi}}{\arg\min}\ \frac{1}{2}\|(\boldsymbol{Y} - \boldsymbol{L}\boldsymbol{R}')\boldsymbol{\Pi}^{-1/2}\|_{\mathrm{F}}^2 + \frac{D}{2}\log|\boldsymbol{\Pi}|. \tag{4.17}$$

This comes from a modified model of (4.1) described by

$$\boldsymbol{y}_i = \boldsymbol{L}\boldsymbol{r}_i + \boldsymbol{\epsilon}_i, \quad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\boldsymbol{0}, \nu_i \boldsymbol{I}) \tag{4.18}$$

where $\boldsymbol{r}_i$ denotes the ith column of $\boldsymbol{R}$. Sec. 4.4 combines ideas from (4.12) and (4.17) to tackle the union-of-subspaces setting.

4.4 Proposed Subspace Clustering Method

For notational simplicity, let $\boldsymbol{Y}_k = \boldsymbol{Y}_{\mathcal{C}_k} \in \mathbb{R}^{D \times N_k}$ denote the submatrix of $\boldsymbol{Y}$ having columns corresponding to data samples that are estimated to belong in the kth subspace, i.e., $\boldsymbol{Y}_k = \boldsymbol{Y}_{\mathcal{C}_k} =$ matrix$(\{\boldsymbol{y}_i \,:\, c_i = k\})$, where $\mathcal{C}_k = \{i \,:\, c_i = k\}$ must be determined. This notation applies similarly to other matrices such as $\boldsymbol{\Pi}_k = \boldsymbol{\Pi}_{\mathcal{C}_k} = \text{diag}(\{\nu_i \,:\, c_i = k\})$. For the union of subspace measurement model in (4.2), we generalize LR-ALPCAH (4.17) by proposing the following cost function

$$\underset{\mathcal{L},\mathcal{R},\boldsymbol{\Pi},\mathcal{C}}{\arg\min} \underbrace{\sum_{k=1}^{K} \overbrace{\frac{1}{2}\left\|(\boldsymbol{Y}_k - \boldsymbol{L}_k\boldsymbol{R}_k')\boldsymbol{\Pi}_k^{-1/2}\right\|_{\text{F}}^2 + \frac{D}{2}\log|\boldsymbol{\Pi}_k|}^{f_k(\boldsymbol{L}_k,\boldsymbol{R}_k,\boldsymbol{\Pi}_k;\boldsymbol{Y}_k)}}_{f(\mathcal{L},\mathcal{R},\boldsymbol{\Pi},\mathcal{C})} \tag{4.19}$$

where $\mathcal{C}, \boldsymbol{\Pi}, \mathcal{L}, \mathcal{R}$ denote the sets of estimated clusters, noise variances, and factorized matrices respectively for each cluster $k = 1, \ldots, K$. Specifically, $\mathcal{L} \triangleq \{\boldsymbol{L}_1, \ldots, \boldsymbol{L}_K\}$ and $\mathcal{R} \triangleq \{\boldsymbol{R}_1, \ldots, \boldsymbol{R}_K\}$. Our algorithm for solving (4.19) is called ALPCAHUS (**ALPCAH** for **U**nion of **S**ubspaces). We solve (4.19) via alternating minimization, and the method alternates between subspace basis estimation and data sample reassignment/clustering. We begin with subspace basis estimation.

Firstly, with data sample assignments fixed, $\forall k \in \{1, \ldots, K\}$ we estimate subspace bases by applying T_1 iterations of alternating minimization by

$$\begin{aligned}\left(\boldsymbol{L}_k^{(T_1)}, \boldsymbol{R}_k^{(T_1)}, \boldsymbol{\Pi}_k^{(T_1)}\right) = \underset{\boldsymbol{L}_k,\, \boldsymbol{R}_k,\, \boldsymbol{\Pi}_k}{\arg\min}\ & f_k(\boldsymbol{L}_k, \boldsymbol{R}_k, \boldsymbol{\Pi}_k; \boldsymbol{Y}_k) \\ & \text{s.t. } \boldsymbol{\Pi}_k \succeq \alpha \boldsymbol{I}.\end{aligned} \tag{4.20}$$

The solution to (4.20) is described in Ref. [13]. However, for completeness and notational consistency, the updates are included in (4.21), (4.22), and (4.24). Given the current t_1 iteration, the updates for the $t_1 + 1$ iteration are given as

$$\begin{aligned}\boldsymbol{L}_k^{(t_1+1)} &= \underset{\boldsymbol{L}_k}{\arg\min}\, f_k\left(\boldsymbol{L}_k, \boldsymbol{R}_k^{(t_1)}, \boldsymbol{\Pi}_k^{(t_1)}; \boldsymbol{Y}_k^{(t_2)}\right) \\ &= \boldsymbol{Y}_k^{(t_2)}\left(\boldsymbol{\Pi}_k^{(t_1)}\right)^{-1}\boldsymbol{R}_k^{(t_1)}\left(\left(\boldsymbol{R}_k^{(t_1)}\right)'\left(\boldsymbol{\Pi}_k^{(t_1)}\right)^{-1}\boldsymbol{R}_k^{(t_1)}\right)^{-1} \\ \boldsymbol{R}_k^{(t_1+1)} &= \underset{\boldsymbol{R}_k}{\arg\min}\, f_k\left(\boldsymbol{L}_k^{(t_1+1)}, \boldsymbol{R}_k, \boldsymbol{\Pi}_k^{(t_1)}; \boldsymbol{Y}_k^{(t_2)}\right)\end{aligned} \tag{4.21}$$

$$= \left(\boldsymbol{Y}_k^{(t_2)}\right)' \boldsymbol{L}_k^{(t_1+1)} \left(\left(\boldsymbol{L}_k^{(t_1+1)}\right)' \boldsymbol{L}_k^{(t_1+1)}\right)^{-1}. \tag{4.22}$$

Further, estimate the noise variance terms by

$$\underbrace{\max\left(\alpha, \frac{1}{|D|}\left\|\left(\boldsymbol{Y}_k^{(t_2)} - \boldsymbol{L}_k^{(t_1+1)}\left(\boldsymbol{R}_k^{(t_1+1)}\right)'\right)\boldsymbol{e}_i\right\|_2^2\right)}_{p_k(i)} \tag{4.23}$$

where $\boldsymbol{e}_i$ denotes the ith canonical basis vector that is used to select the ith residual column and $\alpha > 0$ is a user-selected noise variance threshold parameter. Then, we update the noise variance matrix $\boldsymbol{\Pi}_k$ as follows

$$\begin{aligned}\boldsymbol{\Pi}_k^{(t_1+1)} &= \arg\min_{\boldsymbol{\Pi}_k} f_k(\boldsymbol{L}_k^{(t_1+1)}, \boldsymbol{R}_k^{(t_1+1)}, \boldsymbol{\Pi}_k; \boldsymbol{Y}_k^{(t_2)}) \\ &= \text{diag}(p_k(1), \ldots, p_k(|\mathcal{C}_k|)).\end{aligned} \tag{4.24}$$

Because $\boldsymbol{\Pi}_k$ is a diagonal matrix, the $\boldsymbol{\Pi}_k \succeq \alpha \boldsymbol{I}$ majorization condition in (4.20) being equivalent to $\forall i, \lambda_i(\boldsymbol{\Pi}_k) \geq \alpha$ implies that $\forall i, \nu_i > \alpha$. This leads to a projection to the positive set $[\alpha, \infty)$ by the $\max(\alpha, \cdot)$ condition in (4.23). This condition is sufficient to ensure convergence as proven in [13, Thm. 2] and used in this work to further prove ALPCAHUS convergence in Thm. 3 using some $\alpha \in \mathbb{R}_+$.

Secondly, fixing the subspaces, we update the data sample assignments to clusters. The cluster update is essentially $\mathcal{C}^{\text{new}} = \arg\min_{\mathcal{C}} f(\mathcal{L}, \mathcal{R}, \boldsymbol{\Pi}, \mathcal{C})$, with a rule to break ties in favor of the previous cluster assignment as follows. Let $\boldsymbol{U}_k^{(T_1)}$ denote the Gram-Schmidt vectors of $\boldsymbol{L}_k^{(T_1)}$. Define residual point error for kth basis by

$$J_i(k) = \left\|\boldsymbol{y}_i - \boldsymbol{U}_k^{(T_1)}\left(\boldsymbol{U}_k^{(T_1)}\right)' \boldsymbol{y}_i\right\|_2^2 \tag{4.25}$$

and let the set S_{J_i} denote the set of minimizers of $J_i(k)$ by

$$\mathcal{S}_{J_i} = \arg\min_k J_i(k). \tag{4.26}$$

Then $\forall i$, given the label estimate $c_i^{(t_2)}$ for the t_2 iteration, compute the next label by

$$c_i^{(t_2+1)} = \begin{cases} c_i^{(t_2)} & \text{if } c_i^{(t_2)} \in \mathcal{S}_{J_i} \\ k^* \in \mathcal{S}_{J_i} & \text{otherwise.} \end{cases} \tag{4.27}$$

The solution to (4.27) involves the following procedure. Given a data sample $\boldsymbol{y}_i$, find the lowest subspace projection residual out of all subspaces by (4.25) and assign it to $c_i^{(t_2+1)}$ at the t_2+1 iteration. In the event of a tie, meaning there is more than one subspace that has equal residual, retain the past label $c_i^{(t_2)}$ for the new label $c_i^{(t_2+1)}$. By doing this, cycling between labels is prevented for all data samples. This cluster reassignment criterion will be important for ensuring convergence, as shown in Thm. 3. Because (4.26) is with respect to k and not the ith sample, a residual weighted with the noise parameter ν_i is unnecessary, since it would not change the minimizing k.

To further improve clustering performance, consensus clustering can be leveraged over many trials. Initially, we tried using this approach with only one trial as seen in Fig. 4.2b and ALPCAHUS ($B = 1$) result in Fig. 4.7a. However, since K-subspaces in general is sensitive to initialization, during experimentation, we found higher clustering accuracy by using a consensus approach with more than one trial as shown in Fig. 4.2d and ALPCAHUS ($B = 16$) result in Fig. 4.7a.

4.4.1 Ensemble Extension for ALPCAHUS

This section presents the ensemble algorithm with base clustering parameter B to combine multiple trials for finding $\mathcal{C}$ in (4.19). ALPCAHUS with $B > 1$ leverages consensus clustering by forming an affinity matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ where

$$\begin{aligned}\boldsymbol{A}_{i,j} = &\frac{1}{B}|\{\forall b \in \{1,\dots,B\} \text{ such that} \\ &\boldsymbol{y}_i, \boldsymbol{y}_j \text{ are co-clustered in } \mathcal{C}^{(b)}\}|\end{aligned} \tag{4.28}$$

and $\mathcal{C}^{(b)} = \{c_1^{(b)},\dots,c_N^{(b)}\}$ refers to the cluster labels for the bth trial ranging from 1 to B. Then, the rows and columns of $\boldsymbol{W}$ are thresholded to retain the top q values by solving

$$\boldsymbol{Z}^{\text{row}} = \arg\min_{\boldsymbol{Z}} \|\boldsymbol{A}-\boldsymbol{Z}\|_{\text{F}}^2 \;\text{ s.t. }\; \|\boldsymbol{Z}_{i,:}\|_0 = q \;\; \forall i \tag{4.29}$$

$$\boldsymbol{Z}^{\text{col}} = \arg\min_{\boldsymbol{Z}} \|\boldsymbol{A}-\boldsymbol{Z}\|_{\text{F}}^2 \;\text{ s.t. }\; \|\boldsymbol{Z}_{:,i}\|_0 = q \;\; \forall i. \tag{4.30}$$

Finally, spectral clustering is applied to $\boldsymbol{W} = \frac{1}{2}(\boldsymbol{Z}^{\text{col}} + \boldsymbol{Z}^{\text{row}})$ to obtain the clusters from the ensemble results. This spectral clustering operation assumes a balanced cluster distribution that may not hold in practice. All of our included experiments contain balanced cluster sizes; however, Sec. 4.7.2 provides discussion of this limitation and possible remediation strategies.

The q threshold parameter can be set via cross-validation, as done in the experiments shown in Sec. 4.5, or other techniques can be used. For example, Ref. [122] creates a sparse L2 graph

through hard thresholding to retain the top $q = d_k$ values where d_k is the known subspace dimension. Thus, q can be tied to the true or estimated subspace dimension instead. Sec. 4.7.3 provides more discussion on ALPCAHUS hyperparameters, including recommended values.

In this more general ensemble framework, one could select $B = 1$ to reduce to one trial of the optimization problem in (4.19), which is the non-ensemble version of ALPCAHUS. Alg. ALPCAHUS summarizes the procedure for subpace clustering with optional ensemble learning. For Julia code implementations, refer to github.com/javiersc1/ALPCAHUS.

Algorithm ALPCAHUS (unknown noise variances, unknown noise grouping, ensemble version with random init.)

Input: $\boldsymbol{Y} \in \mathbb{R}^{D\times N}$: data, $K \in \mathbb{Z}^+$: number of subspaces, $\{\hat{d}_k \in \mathbb{Z}^+ \ \forall k \in \{1,\dots,K\}\}$: candidate dimension for all clusters, $q \in \mathbb{Z}^+$: threshold parameter, $B \in \mathbb{Z}^+$: base clusterings, $T_1 \in \mathbb{Z}^+$: LR-ALPCAH iterations, $T_2 \in \mathbb{Z}^+$: maximum alternating updates (cluster reassignments)
Output: $\mathcal{C}_S = \{c_1,\dots,c_N\}$: clusters of $\boldsymbol{Y}$
for $b = 1,\dots,B$ (in parallel) **do**
  $\mathcal{C}_k \sim \{1,\dots,N\}$ s.t. $|\mathcal{C}_k| \approx \frac{N}{K}$ for $k = 1,\dots,K$ — Initialize clusters randomly
  $\boldsymbol{\Pi}_k \leftarrow \boldsymbol{I}$ for $k = 1,\dots,K$ — Assume homoscedastic data initially
  $\boldsymbol{L}_k, \boldsymbol{R}_k \leftarrow \boldsymbol{U}_{:,1:\hat{d}_k} \boldsymbol{\Sigma}^{1/2}_{1:\hat{d}_k,1:\hat{d}_k}, \ \boldsymbol{\Sigma}^{1/2}_{1:\hat{d}_k,1:\hat{d}_k} \boldsymbol{V}_{:,1:\hat{d}_k}$ where $\text{SVD}(\boldsymbol{Y}_{\mathcal{C}_k}) = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}'$ for $k = 1,\dots,K$ — Spectral init
  for $t_2 = 1,\dots,T_2$ (in sequence) **do**
    $\boldsymbol{L}_k, \boldsymbol{R}_k, \boldsymbol{\Pi}_k \leftarrow$ Compute (4.21)-(4.24) using $\boldsymbol{Y}_{\mathcal{C}_k}$ for T_1 iterations for $k = 1,\dots,K$ — Apply LR-ALPCAH [13]
    $\mathcal{C}_k \leftarrow \{\forall \boldsymbol{y}_i \in \boldsymbol{Y} : \forall k$, apply (4.25)-(4.27) with $\boldsymbol{U}_k$ from GramSchmidt$(\boldsymbol{L}_k)\}$ — Update cluster labels
    Stop execution if $\mathcal{C}_k^{(t_2+1)} = \mathcal{C}_k^{(t_2)}$ is met for all clusters — Stopping criteria
  end for
  $\mathcal{C}^{(b)} \leftarrow \mathcal{C} = \{c_1,\dots,c_N\}$ — Collect results from all trials
end for
$\boldsymbol{A}_{i,j} \leftarrow$ Form $\boldsymbol{A}$ by (4.28) using $\{\mathcal{C}^{(1)},\dots,\mathcal{C}^{(B)}\}$ — Form affinity matrix of similar clusterings
$\boldsymbol{Z}^{\text{row}} \leftarrow$ Threshold rows by solving (4.29) using $\boldsymbol{A}$ — Retain top q elements for each row
$\boldsymbol{Z}^{\text{col}} \leftarrow$ Threshold columns by solving (4.30) using $\boldsymbol{A}$ — Retain top q elements for each column
$\boldsymbol{W} \leftarrow \frac{1}{2}\left(\boldsymbol{Z}^{\text{row}} + \boldsymbol{Z}^{\text{col}}\right)$ — Average affinity matrix
$\mathcal{C}_S \leftarrow$ Perform spectral clustering on $\boldsymbol{W}$ via (4.4)-(4.7) — Final clustering

4.4.2 ALPCAHUS Convergence

In the single cluster setting, ALPCAHUS with $K = 1$ has a sequence of cost function values $f_1(\boldsymbol{L}_1, \boldsymbol{R}_1, \boldsymbol{\Pi}_1; \boldsymbol{Y}_1)$ that provably converges to local minima, since the cost function and algorithm simplify to LR-ALPCAH (4.17) that has convergence guarantees as proven in Ref. [13, Thm. 2].

In the multi-cluster setting with $K > 1$, Thm. 3 below shows that the sequence of cost function values produced by $f(\mathcal{L}, \mathcal{R}, \mathbf{\Pi}, \mathcal{C})$ in (4.19) converges asymptotically. The argument in Thm. 3 below is a natural extension of that in the original k-subspaces paper [116, Thm. 7] with greater mathematical exposé. Sec. 4.7.1 provides an experimental result that corroborates this theorem.

Theorem 3. *Consider the ALPCAHUS cost function $f(\mathcal{L}, \mathcal{R}, \mathbf{\Pi}, \mathcal{C})$ in (4.19). Assume a noise variance threshold parameter $\alpha \in \mathbb{R} > 0$ that lower bounds all ν_i, and the cluster assignment criteria in (4.27) that accepts changes only if there is a cluster reassignment of points that strictly decreases the cost function $f(\cdot)$, as expressed in (4.27). Then, (4.19) generates a sequence of cost function values that converges asymptotically. Furthermore, the algorithm terminates in finite iterations due to stopping criteria in line* (10) *of Alg. ALPCAHUS that checks whether clusters have changed from the previous iteration.*

Proof. To prove that the sequence of cost function values converges, we show that each step decreases the cost function. We also show that the algorithm terminates when the variables stop changing. Each step consists of two sub-steps: first, the subspace estimation sub-step, where LR-ALPCAH is used, and second, the clustering sub-step where projection onto the estimated subspaces is done. Let t_1 be the iteration variable associated with the variables in the subspace estimation sub-step and t_2 be the iteration variable associated with the variables in the clustering sub-step. Define $\boldsymbol{X}_k = \boldsymbol{L}_k \boldsymbol{R}_k'$ and bound the cost by

$$
\begin{aligned}
& f(\mathcal{L}^{(t_1)}, \mathcal{R}^{(t_1)}, \mathbf{\Pi}^{(t_1)}, \mathcal{C}^{(t_2)}) = \\
& \frac{1}{2} \sum_k \sum_{i \in \mathcal{C}_k^{(t_2)}} \frac{1}{\nu_i^{(t_1)}} \left\| [\boldsymbol{Y}_k]_i - [\boldsymbol{X}_k]_i^{(t_1)} \right\|_2^2 + D \log \nu_i^{(t_1)} && (4.31) \\
& \geq \frac{1}{2} \sum_k \sum_{i \in \mathcal{C}_k^{(t_2)}} \frac{1}{\nu_i^{(t_1+1)}} \left\| [\boldsymbol{Y}_k]_i - [\boldsymbol{X}_k]_i^{(t_1+1)} \right\|_2^2 \\
& + D \log \nu_i^{(t_1+1)} = f(\mathcal{L}^{(t_1+1)}, \mathcal{R}^{(t_1+1)}, \mathbf{\Pi}^{(t_1+1)}, \mathcal{C}^{(t_2)}) && (4.32) \\
& \geq \frac{1}{2} \sum_k \sum_{i \in \mathcal{C}_k^{(t_2+1)}} \frac{1}{\nu_i^{(t_1+1)}} \left\| [\boldsymbol{Y}_k]_i - [\boldsymbol{X}_k]_i^{(t_1+1)} \right\|_2^2 \\
& + D \log \nu_i^{(t_1+1)} = f(\mathcal{L}^{(t_1+1)}, \mathcal{R}^{(t_1+1)}, \mathbf{\Pi}^{(t_1+1)}, \mathcal{C}^{(t_2+1)}). && (4.33)
\end{aligned}
$$

Here, (4.31) $\geq$ (4.32) is due to the subspace step not increasing the cost for each f_k term as shown in Ref. [13, Thm. 2], and (4.32) $\geq$ (4.33) is due to the cluster reassignment criteria in (4.27) only updating the cluster label that decreases (4.25). What has been shown is that

$$
f^{(t_1,t_2)} \geq f^{(t_1+1,t_2)} \geq f^{(t_1+1,t_2+1)} \tag{4.34}
$$

where $f^{(t_1,t_2)} = f(\mathcal{L}^{(t_1)}, \mathcal{R}^{(t_1)}, \mathbf{\Pi}^{(t_1)}, \mathcal{C}^{(t_2)})$. Thus, as the cluster assignments and subspace bases are updated, the cost is non-increasing. Since the cost function values are non-increasing at every step, bounded below, then the sequence of cost function values produced by (4.19) converges asymptotically. Furthermore, the stopping criteria in line (10) ensures that the algorithm terminates when all clusters do not change. Because there are only a finite number of ways to assign cluster labels, Alg. ALPCAHUS will stop after a finite number of iterations. □

4.4.3 Rank Estimation

In subspace clustering, some algorithms, like SSC, do not require the subspace dimension to be known or estimated, whereas others like KSS require it. For the group of algorithms that require this parameter, there is great interest in adaptive methods that can learn the subspace dimension. In recent work, Ref. [117] proposed using an eigen-gap heuristic on the sample covariance matrix of each cluster to estimate dimension. Mathematically, this means calculating the following

$$\boldsymbol{S}_k = \frac{1}{|\mathcal{C}_k|}\boldsymbol{Y}_k\boldsymbol{Y}_k' \tag{4.35}$$

$$\hat{d}_k = \arg\max_i |\lambda_i(\boldsymbol{S}_k) - \lambda_{i+1}(\boldsymbol{S}_k)| \tag{4.36}$$

where $\lambda_i(S_k)$ denotes the ith eigenvalue of S_k assuming $\lambda_1 \geq \ldots \geq \lambda_D$. For heteroscedastic data, the eigen-gap heuristic can break down, making it challenging to determine rank, as shown in Sec. 4.5, Fig. 4.5. In recent work, Ref. [79] developed a parallel analysis algorithm to estimate the rank of a matrix that is consistently shown to work well in the heteroscedastic regime if the data comes from a *single* subspace only. It works by creating an i.i.d. Bernoulli $(p = 0.5)$ matrix denoted as $\boldsymbol{M}$ and analyzing the singular values of $\boldsymbol{M} \odot \boldsymbol{Y}$ for a matrix of data samples $\boldsymbol{Y}$. This process allows one to distinguish which singular values are associated with the signal and noise components of the data. We generalize this line of work and apply it to the union of subspace setting by starting off over-parameterized and adaptively shrinking the bases by

$$\begin{aligned} &\forall k \in \{1, \ldots, K\} \text{ do} \\ &\tilde{\sigma}^{(r)} = \text{SingularValues}(\boldsymbol{M} \odot \boldsymbol{Y}_k) \ \ \forall r \in \{1, \ldots, R\} \qquad (4.37) \\ &\hat{d}_k = \text{smallest } d \text{ that satisfies} \\ &\quad \sigma_{d+1}(\boldsymbol{Y}_k) \leq \alpha\text{-percentile of } \{\tilde{\sigma}_{d+1}^{(1)}, \ldots, \tilde{\sigma}_{d+1}^{(R)}\}. \qquad (4.38) \end{aligned}$$

This is done for each estimated cluster $\boldsymbol{Y}_k$ over R random trials where $1 \leq R \ll T_2$, and T_2 denotes the maximum ALPCAHUS iterations. Here, $\sigma_{d+1}(\boldsymbol{Y}_k)$ denotes the $d+1$ singular value of $\boldsymbol{Y}_k$. This process is repeated for all cluster subsets $\{\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_K\}$ after the cluster reassignment

update in (4.27). To reduce computation, this dimension estimation is performed sparingly in the ALPCAHUS method.

4.4.4 Cluster Initialization

For the non-ensemble version of ALPCAHUS ($B = 1$), it previously remained to be seen whether there exists an initialization scheme that performs better than random cluster assignment in the heteroscedastic regime. In recent work, Ref. [117] proposed a thresholding inner-product based spectral initialization method (TIPS), designed for homoscedastic data to be used with KSS, where an affinity matrix is generated by $\boldsymbol{W}_{ij} = 1$ if $|\langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle| \geq \tau$ and $i \neq j$ given a thresholding parameter $\tau > 0$. Instead, in this work, we generate a fully connected $\boldsymbol{W}$ by creating

$$\boldsymbol{W}_{ij} = |\langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle| \text{ if } i \neq j,\ 0 \text{ otherwise,} \tag{4.39}$$

and hard threshold to retain the top q edges by solving (4.11) so that $\tau = q$ for simplicity. Then, the cluster assignments $\mathcal{C} = \{c_1, \ldots, c_N\}$ are calculated by applying the spectral clustering method on the thresholded $\boldsymbol{W}$. Recall that $\boldsymbol{y}_i = \boldsymbol{x}_i + \boldsymbol{\epsilon}_i$. Upon closer analysis, this metric in expectation, ignoring absolute value, gives

$$\begin{aligned}\mathbb{E}[\langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle] &= \mathbb{E}[\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle] + \mathbb{E}[\langle \boldsymbol{x}_i, \boldsymbol{\epsilon}_j \rangle] \\ &\quad + \mathbb{E}[\langle \boldsymbol{\epsilon}_i, \boldsymbol{x}_j \rangle] + \mathbb{E}[\langle \boldsymbol{\epsilon}_i, \boldsymbol{\epsilon}_j \rangle] = \mathbb{E}[\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle].\end{aligned} \tag{4.40}$$

Therefore, the metric is independent of the noise variances, meaning the affinity matrix constructed does not have highly unbalanced, asymmetric edge weights from noisy samples. Thus, this metric is more robust to heteroscedastic noise than others such as Euclidean norm, where

$$\mathbb{E}[\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2] = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 + D(\boldsymbol{\nu}_i + \boldsymbol{\nu}_j). \tag{4.41}$$

Clearly, this is inflated by the noise terms $\boldsymbol{\nu}_i$ and $\boldsymbol{\nu}_j$. To note, TIPS initialization is only useful when $B = 1$ for one base clustering since it is a deterministic initialization; it provably performs better than random initialization as illustrated in Fig. 4.4. Otherwise, random initialization is used when $B > 1$ to leverage consensus information in the ensemble process.

4.5 Experiments

4.5.1 Synthetic Experiments

4.5.1.1 Experimental Setup

A synthetic dataset is generated, consisting of $K = 2$ clusters, each of dimension $d = 3$ derived from random subspaces in $D = 100$ dimensional ambient space. Each cluster consisted of two data groups with group 1 containing $N_1 = 6$ samples, to explore the data constrained regime, with noise variance $\nu_1 = 0.1$ per cluster. For group 2, the N_2 samples and noise variance ν_2 are varied. Cross-validation is performed for hyperparameters in any algorithm that requires it, using a separate training set. For a more detailed description of ALPCAHUS parameters, including recommended default values, see Sec. 4.7.3. Both the estimated clusters from each clustering algorithm and the known clusters are necessary to calculate clustering error. Further, the problem of label permutations is addressed using the Hungarian algorithm [123]. More concretely, clustering error (%) is defined as

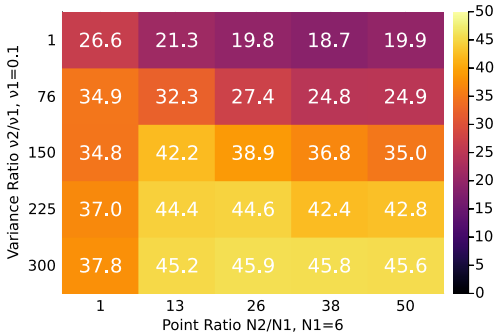
$$\text{clustering error} = \frac{100}{N}\left(1 - \max_{\pi} \sum_{i,j} \boldsymbol{Q}^{\text{out}}_{\pi(ij)} \boldsymbol{Q}^{\text{true}}_{ij}\right) \tag{4.42}$$

where π is a permutation of cluster labels found by applying the Hungarian algorithm, and $\boldsymbol{Q}^{\text{out}}$ and $\boldsymbol{Q}^{\text{true}}$ are output labels and ground truth labelings of the data where the (i, j)th entry is one if point j belongs to cluster i and zero otherwise.

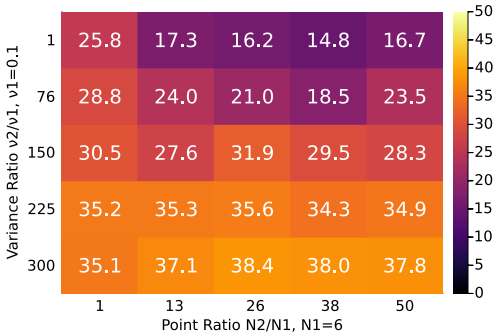
The average clustering error is computed across 100 trials, where each trial used different noise, basis coefficients, and subspace basis realizations for the clusters. Various subspace clustering algorithms are included such as K-Subspaces (KSS) [24], Ensemble K-Subspaces (EKSS) [108], Subspace Clustering via Thresholding (TSC) [107], and Doubly Stochastic Sparse Subspace Clustering (ADSSC) [124], which is a modern variant of SSC. Applying general non-subspace clustering methods such as K-means [20] yielded poor clustering performance, so the results of these methods are not shown in general.

4.5.1.2 Clustering Error

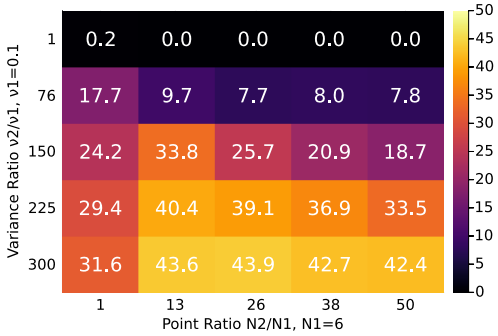
Table 4.2h shows the clustering quality of these algorithms for synthetic heteroscedastic data. Fig. 4.2 complements Table 4.2h by exploring the complete heteroscedastic landscape with a visual representation that is easier to understand. We define a method called "noisy oracle" where an oracle uses the true cluster assignments to apply PCA only to the low-noise data for each cluster. Using these subspace basis estimates of each cluster, the oracle performs cluster assignments using



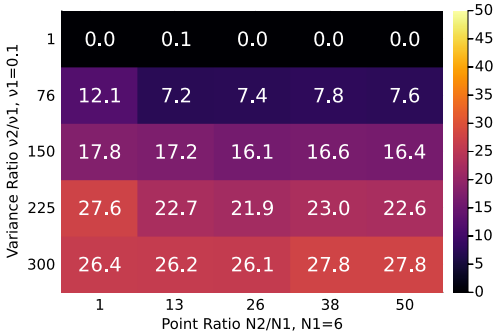
(a) KSS (TIPS) mean clustering error.



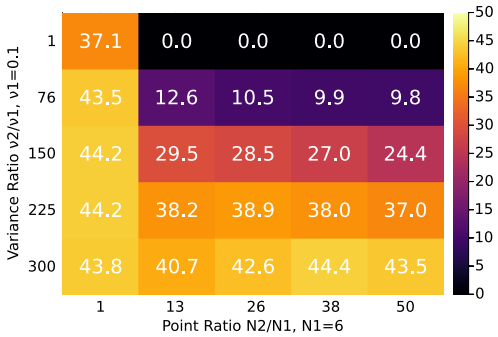
(b) ALPCAHUS ($B = 1$, TIPS) mean clustering error.



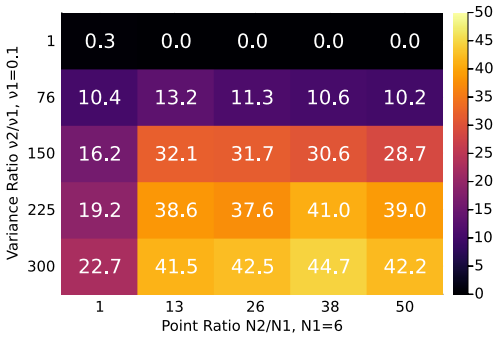
(c) EKSS (B=128) mean clustering error.



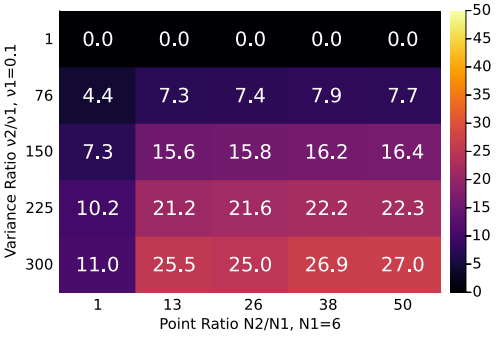
(d) ALPCAHUS (B=128) mean clustering error.



(e) TSC mean clustering error.



(f) ADSSC mean clustering error.



(g) Noisy oracle mean clustering error.

ν_2/ν_1	1	1	300	300	150	225	76
N_2/N_1	1	50	1	50	26	13	38
Reference Level							
Noisy Oracle	0.0	0.0	11.0	27.0	15.8	21.2	7.9
Method Comparisons							
KSS	26.6	19.9	37.8	45.6	38.9	44.4	24.8
EKSS ($B = 128$)	0.2	**0.0**	31.6	42.4	25.7	40.4	8.0
ADSSC	0.3	**0.0**	**22.7**	42.2	31.7	38.6	10.6
TSC	37.1	**0.0**	43.8	43.5	28.5	38.2	9.9
ALPCAHUS ($B = 1$)	25.8	16.7	35.1	37.8	31.9	35.3	18.5
ALPCAHUS ($B = 128$)	**0.0**	**0.0**	26.4	**27.8**	**16.1**	**22.7**	**7.8**

(h) Clustering error (%) results.

Figure 4.2: Clustering error over the heteroscedastic landscape for subspace clustering algorithms.

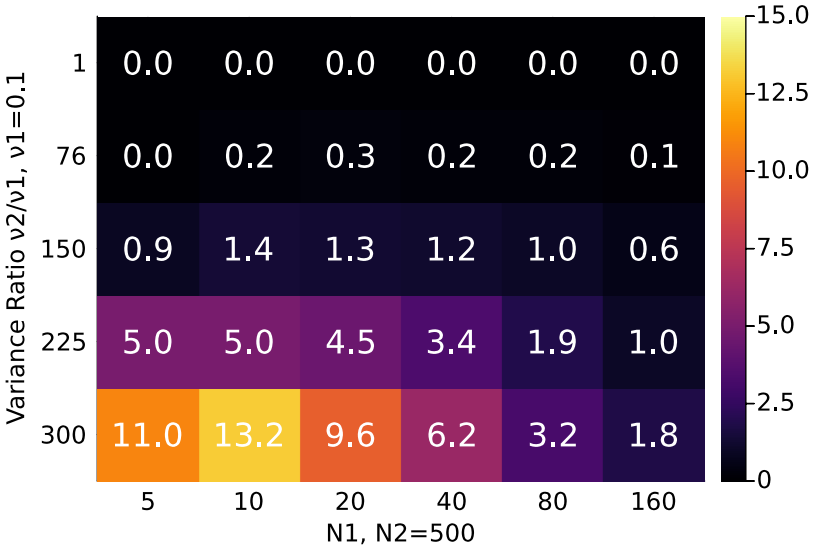


Figure 4.3: Percentage difference (%) of ALPCAHUS clustering error subtracted from EKSS while the amount of good data varies.

(4.27). This oracle reference provides a kind of lower bound on realistic clustering performance since it can approximate subspace bases separately, given the true labeling.

ALPCAHUS ($B = 1$) with TIPS initialization achieved a lower clustering error than KSS with TIPS initialization. For the ensemble methods ($B > 1$), both EKSS and ALPCAHUS significantly improved. However, EKSS still had higher clustering error in more heteroscedastic regions. Meanwhile, ALPCAHUS remained very close to the noisy oracle, indicating it more accurately estimated the underlying true labeling. Compared to other methods like TSC and ADSSC, ALPCAHUS generally outperformed them. Overall, the ensemble version of ALPCAHUS was generally more robust against heteroscedasticity compared to other subspace clustering algorithms. To summarize, the effects of data quality and data quantity are explored on the heteroscedastic landscape. To our knowledge, this is the first systematic analysis of heteroscedasticity effects on subspace clustering quality in the literature.

4.5.1.3 Effects of Good Data

Additionally, the effects of good data quantity on the subspace clustering problem is explored to understand at what point it becomes advantageous to use ALPCAHUS over KSS methods. The following parameters are fixed $\{\nu_1 = 0.1, N_2 = 500\}$ and the following vary $\{N_1, \nu_2\}$. The reporting metric is the percentage difference of EKSS clustering error (%) - ALPCAHUS clustering error (%), i.e., $\text{error}_{\text{EKSS}} - \text{error}_{\text{ALPCAHUS}}$, meaning higher values indicate that ALPCAHUS is better than EKSS. Fig. 4.3 shows that ALPCAHUS performed better than EKSS even up to $N_1 = 160$ which represents about 33% of the total data. ALPCAHUS never performed worse than EKSS, but

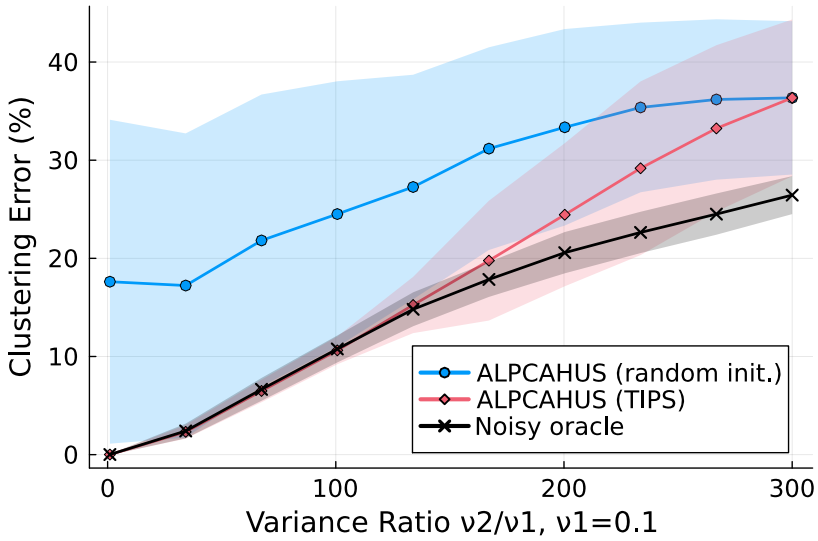


Figure 4.4: Clustering error (%) for TIPS initialization scheme vs. random initialization for the ALPCAHUS method ($B = 1$).

the advantage gap narrowed as N_1 increased. Thus, there is a wide range of conditions for which ALPCAHUS is preferable to homoscedastic methods.

4.5.1.4 Clustering Initialization

Sec. 4.4.4 proposed using the TIPS initialization scheme in the heteroscedastic context since the dot product metric used to construct the affinity matrix in (4.40) is provably robust to heteroscedastic noise. We fixed the following parameters $\{N_1, N_2\}$ and varied ν_2 while keeping $\nu_1 = 0.1$. The noisy oracle result is included to establish a realistic performance baseline. Fig. 4.4 shows that TIPS initialization ALPCAHUS $(B = 1)$ outperformed random initialization for the non-ensemble ALPCAHUS method $(B = 1)$ across the entire heteroscedastic landscape. Thus, TIPS should always be used for the non-ensemble version for higher clustering accuracy. The ensemble version of ALPCAHUS $(B > 1)$ is not included in these results because the ensemble process inherently takes advantage of random initialization to achieve a different labeling for each trial.

4.5.1.5 Rank Estimation of Clusters

In Sec. 4.4.3, a random sign flipping method is proposed to adaptively find and shrink the subspace dimension of clusters when the subspace dimension is unknown. In Fig. 4.5, synthetic subspaces are generated with true rank $d = 6$ to explore how the initial rank parameter affects the ability to estimate the true rank over 100 trials. This approach is compared against the eigen-gap heuristic by using ALPCAHUS with both adaptive rank schemes and reports the estimated rank values from

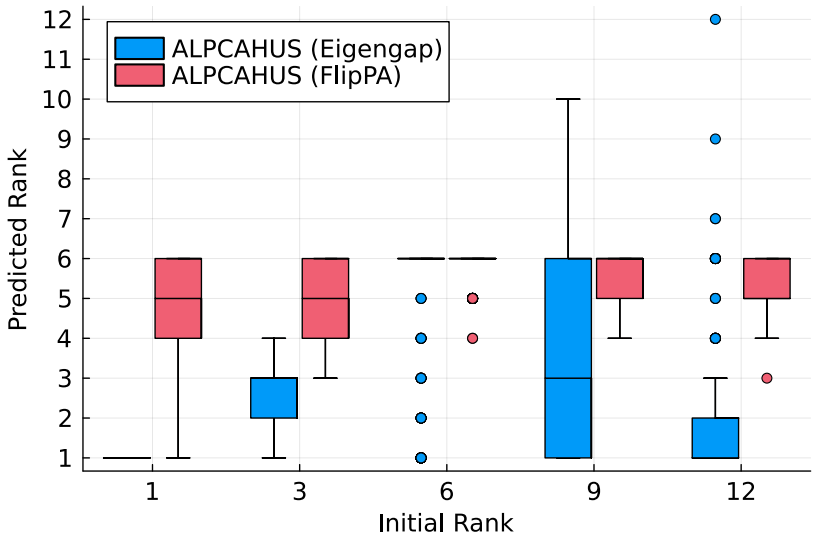


Figure 4.5: Adaptive rank estimation using eigengap heuristic and proposed FlipPA approach (true rank $d = 6$).

Methods	Time (ms)	Memory (MiB)	Mean Subspace Error	Mean Clustering Error (%)
KSS	149.3	53.0	0.80	38.5
EKSS (B=16)	181.4	212.6	0.62	23.8
ADSSC	125.7	17.5	0.62	21.1
TSC	**32.3**	**16.5**	0.40	18.2
ALPCAHUS (B=1)	148.5	126.7	0.43	14.4
ALPCAHUS (B=16)	210.7	678.2	**0.28**	**6.3**

Table 4.1: Subspace clustering results on quasar flux data. The KSS and ALPCAHUS ($B = 1$) methods use TIPS initialization.

the final clustering. The eigen-gap approach consistently underestimated the rank regardless of the initial value, except when given the true rank value $d = 6$. Our sign flipping approach provided much better rank estimates with smaller variances between trials.

4.5.2 Real Data Experiments

4.5.2.1 Quasar Flux Data

Quasar spectra data is downloaded from the Sloan Digital Sky Survey (SDSS) Data Release 16 [92] using the DR16Q quasar catalog [93]. Each quasar has a vector of flux measurements across wavelengths that describes the intensity of observing that particular wavelength. In this dataset, the noise is heteroscedastic across the sample space (quasars) and feature space (wavelength), but for

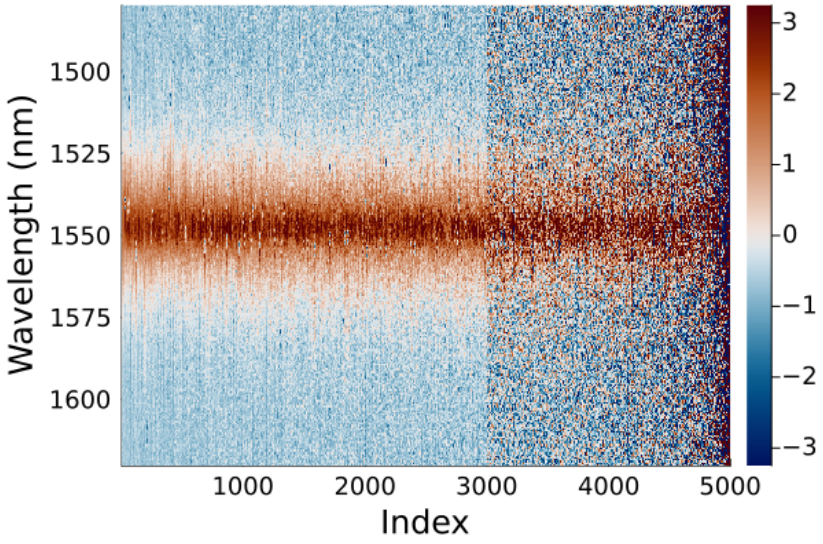
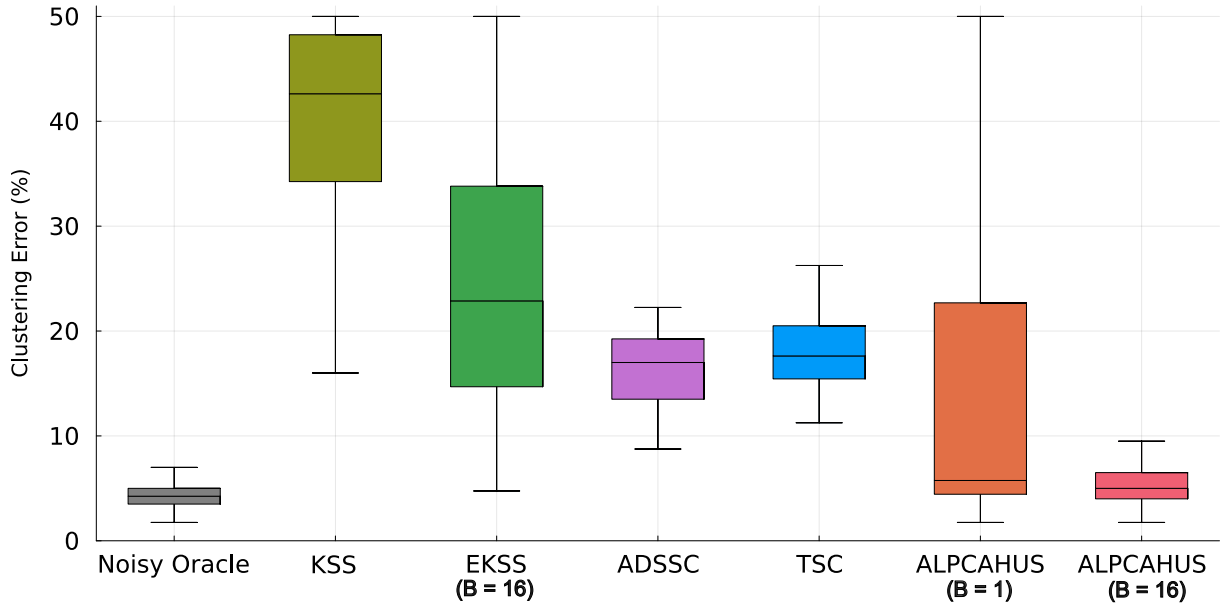


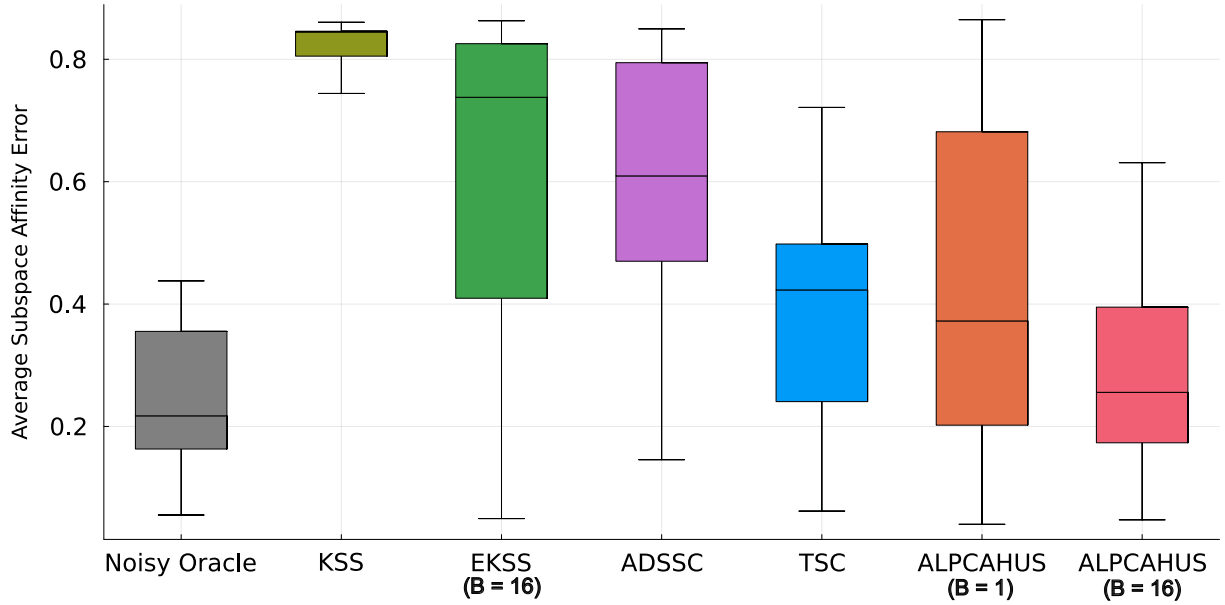
Figure 4.6: Sample data matrix of quasar flux measurements across wavelengths for each column-wise sample.

these experiments, we focused on a subset of data that is homoscedastic across wavelengths and heteroscedastic across quasars. The noise for each quasar is known, given the measurement devices used for data collection [92]. In Fig. 4.6, a subset of the spectra data is shown for illustrative purposes to compare the visual differences in data quality. The preprocessing pipeline for the data included filtering, interpolation, centering, and normalization based on the supplementary material of Ref. [94]. Clusters are formed in these experiments by considering quasars with different properties, namely, two quasar groups ($K = 2$) with different redshift values ($z_1 = 1.0-1.1, z_2 = 2.0-2.1$). Additionally, for the second group, only broad absorption line (BAL) type quasar data is collected. Since the downloaded data is queried separately, clustering association is known, meaning one can compute clustering error for comparison purposes. A training set (5000 samples) is set aside to learn any model parameters, and the rest (5000 samples) is used for a test dataset. Trials are formed by randomly selecting 400 quasar spectra samples per group and running clustering algorithms for 50 total trials. For rank estimation, the noisy oracle approach is used, given the known cluster labels, to find that there is no improvement to clustering quality for rank values greater than $\hat{d} = 3$, so this value is used for any applicable algorithms.

Fig. 4.7a shows clustering errors for this multi-cluster data set of quasar spectra. The ensemble version of ALPCAHUS was very close in clustering quality to the noisy oracle, suggesting that it accurately learned the subspace bases while clustering the data groups. The non-ensemble version of ALPCAHUS ($B = 1$) had large variances in clustering error, indicating some challenges in getting close to the optimal cost function minima with this data. However, based on median values, it still achieved a lower error than KSS, ADSSC, and TSC. Additionally, Fig. 4.7b shows the



(a) Subspace clustering results for various methods.



(b) Average subspace affinity error results for various methods.

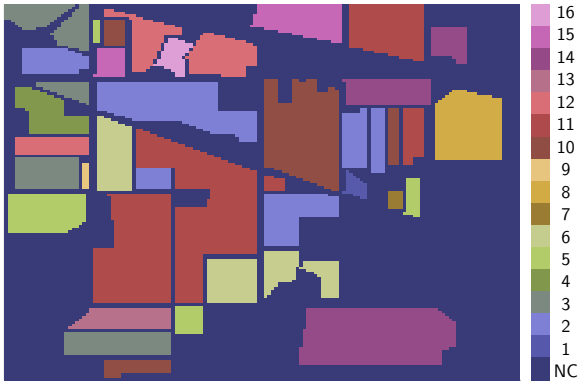
Figure 4.7: Experimental results of quasar flux data for subspace clustering and learning. Methods involving a single run, i.e., KSS and ALPCAHUS ($B = 1$), use the TIPS initialization scheme.

average subspace affinity error of these methods after applying LR-ALPCAH to the clusterings for all methods. ALPCAHUS learned the subspace bases better than the other methods when $B = 16$, the smallest value that had small run-to-run variances without taking significantly longer to compute. Both of these results show the benefit of developing heteroscedastic-based algorithms in the subspace clustering context. Table 4.1 reports median time complexity and memory requirements along with mean clustering error and mean subspace error. In this quasar example, the ensemble method gets close in time to the non-ensemble methods due to our multi-threaded implementation. Yet, because of the multi-threaded implementation, the memory requirements are larger for the ensemble method as opposed to the non-ensemble method. Overall, relative to other clustering methods, ALPCAHUS is competitive in terms of time, but comes at the cost of increased memory requirements.

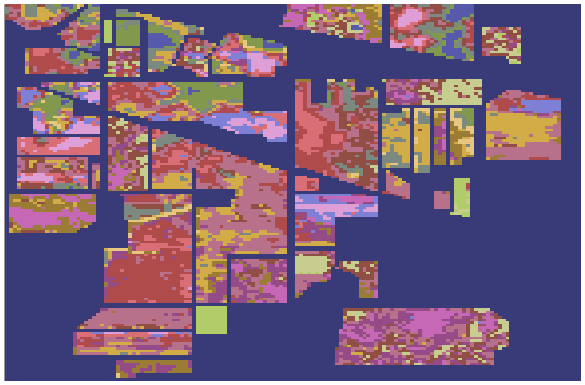
4.5.2.2 Indian Pines Data

Hyperspectral image (HSI) segmentation data is explored in this section, specifically Indian Pines [125]. This image of size 145×145 contains $D = 200$ reflectance bands for each pixel, which is around the $0.4 - 2.5\mu m$ wavelength. HSI data is known to be very noisy due to thermal, atmospheric, and camera-electronics effects, leading to work on per-pixel noise estimation in hyperspectral imaging [126]. In total, there are $K = 16$ classes: alfalfa, corn, grass, wheat, soybean, and others. There is one additional class (background) that is withheld from clustering as commonly done in other works [74]. Because of this, the data contains $N = 10,249$ samples rather than $N = 145 \times 145$. In other work, researchers have found that $\hat{d} = 5$ is an appropriate rank parameter, as it accounts for 95% of the cumulative variance [74]. A subset of the data matrix $Y \in \mathbb{R}^{200 \times 10249}$ is split into 2500 samples to learn model parameters. From this, the learned subspaces are applied to Y to cluster all data, enabling comparison of the heatmaps with the ground truth. The ground truth image is found in Fig. 4.8a. For perspective, random guessing would yield 94% clustering error with $K = 16$ clusters. For consistency, the Hungarian algorithm is again applied to the clustering results in Fig. 4.8 to facilitate comparison with the ground-truth label image in Fig. 4.8a.

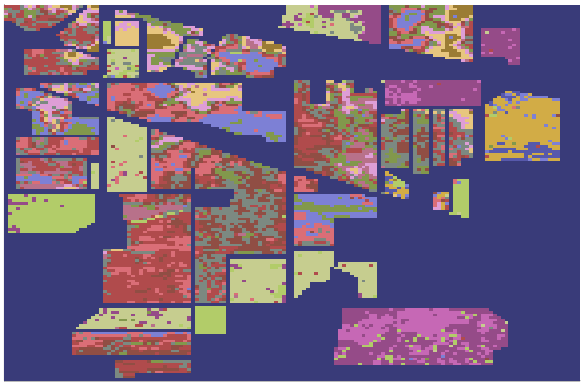
K-means is included in Fig. 4.8b to illustrate the difficulty of the problem. ALPCAHUS results are shown in Fig. 4.8d next to EKSS results in Fig. 4.8c. Clustering error is reported for these algorithms, along with mean intersection over union (IoU) to measure image similarity. ALPCAHUS achieved the lowest clustering error and highest mIOU values (53%, 31%) relative to the other approaches, such as K-means (77%, 16%) and EKSS (64%, 27%). We note that our ALPCAHUS results are similar to those of Ref. [74], even though the authors treat the noise as homoscedastic and the signal as heteroscedastic. Further, the EKSS results on this dataset are similar to the homoscedastic PCA approach used in Ref. [74]. This indicates some utility in



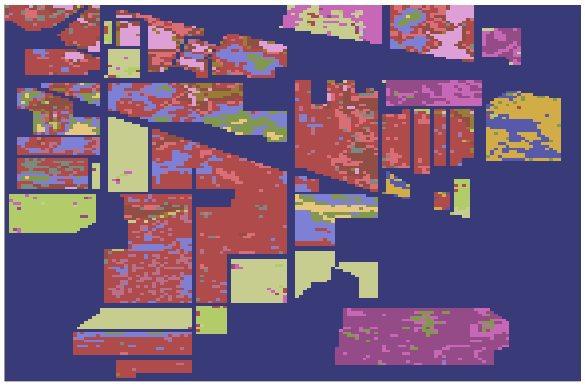
(a) Ground truth labels for *Indian Pines* dataset. NC means no class available (background).



(b) K-means results with clustering error = 77% and mIOU = 16%.



(c) EKSS ($B = 32, T = 3, q = K$) results with clustering error = 64% and mIOU = 27%.



(d) ALPCAHUS ($B = 32, T = 3, q = K$) results with clustering error = 53% and mIOU = 31%.

Figure 4.8: Experimental results of *Indian Pines* HSI data in a subspace clustering context. The results reported are the clustering error and the mean IOU (intersection over union) for each algorithm.

modeling heteroscedasticity in hyperspectral images. Yet, the reflectance bands themselves also appear to be heteroscedastic, with some bands being noisier than others [127]. Thus, developing a method that is doubly heteroscedastic with respect to both the samples and features is an interesting direction for future work.

For reference, the state-of-the-art result is about 10% misclassification rate in a *classification* setting (i.e., not clustering) [128]. In a clustering setting, recent works such as Ref. [129] have achieved a 40% clustering error, which is 13% lower than our results. We do not claim state-of-the-art results with this HSI data, only that the heteroscedastic union of subspace modeling improved results over the homoscedastic union of subspace modeling in finding reflectance band groups with similar characteristics.

4.6 Conclusion

This chapter proposed ALPCAHUS, a subspace clustering algorithm that can find subspace clusters whose samples contain heteroscedastic noise. For future work, a generalization of the union of manifolds by deep learning could be useful. For example, Alzheimer's disease patients often have resting state functional MRI activations different than cognitively normal individuals [130], so one could consider these different classes to belong to different manifolds in the ambient space. Previous research has shown manifold learning to more correctly model temporal dynamics as opposed to subspace modeling in this domain [100]. Another direction for future work could be to consider heteroscedasticity across the feature space rather than the sample space. For example, one might be interested in the biological sequencing of different species with similar genes where the gene marker counts naturally follow heteroscedastic distributions [131]. Another interesting topic would be to explore Thm. 3 in greater detail. This theorem simply states that ALPCAHUS generates a sequence of cost function values that converge; it says nothing about recovery guarantees or the quality of that solution. Thus, proving that the basin of attraction changes as heteroscedastic data is introduced would lead to a greater understanding of heteroscedasticity in a subspace clustering context. Additionally, as noted in Sec. 4.7.2, extending the method to handle extremely imbalanced cluster sizes would be another interesting avenue. These generalizations are nontrivial and are left for future work.

While our implementation of ALPCAHUS benefited from parallelization, it needed more computation time and memory than TSC. Because of this, there is an opportunity to further optimize our code base to remove certain matrix multiplication operations and instead use single vector dot products to reduce memory. Further, since our subspace basis step requires an SVD computation for an initial low-rank estimate, perhaps the Krylov-based Lanczos algorithm [132] can be used to

further reduce time and memory for each trial, leading to more significant improvements in memory usage and time. These additions can be fruitful in big data settings. Overall, ALPCAHUS was more robust to heteroscedastic noise than other clustering methods, making it easier to identify correct clusterings under this heteroscedasticity model.

4.7 Additional Results

4.7.1 ALPCAHUS Convergence Experiment

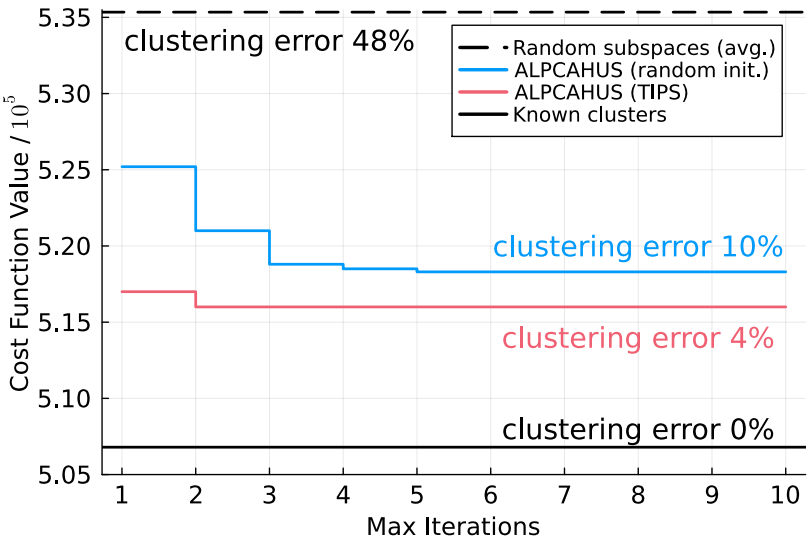


Figure 4.9: ALPCAHUS ($B = 1$) cost function value convergence plot to corroborate the theorem in Thm. 3.

This section focuses on providing empirical verification of the ALPCAHUS convergence theorem in Thm. 3. Quasar spectral flux data from the SDSS (DR16Q catalog) are used in this experiment. The noise variance threshold parameter α is set to 10^{-6} to provide a lower bound for the noise variance estimate. Further, as shown in Thm. 3, it is necessary to use the cluster reassignment criteria in (4.27) that rejects repeated assignments of points to prevent cycling. For simplicity, only one base clustering ($B = 1$) is used in this experiment to compare a randomly initialized ALPCAHUS trial with the TIPS-initialized variant. This variant is only possible when $B = 1$ as discussed in the chapter. In Fig. 4.9, the cost function value is plotted over iterations for both versions of ALPCAHUS.

Both upper and lower bound cost function reference values are provided in Fig. 4.9. The upper bound is generated by using randomly initialized subspaces, and by applying (4.27) to get cluster

labels. From this, the cost function value is computed by (4.19). Likewise, a lower bound is generated by using the known cluster labels, and by estimating the subspaces and noise variances by (4.20). As observed in Fig. 4.9, ALPCAHUS converges relatively quickly with only a few iterations. Note that in this figure, the stopping criteria in line (10) of Alg. ALPCAHUS is purposely ignored. In other words, ALPCAHUS is run for a fixed number of iterations. However, as clearly demonstrated, convergence is achieved earlier for both versions of ALPCAHUS before the maximum number of iterations is reached. This indicates the usefulness of the stopping criteria for speeding up the algorithm across various applications.

4.7.2 Balanced Cluster Assumption

The ensemble extension of ALPCAHUS relies on forming an affinity matrix in (4.28) that is thresholded by (4.29) and (4.30). Then, spectral clustering is applied on said affinity matrix, which depends on (4.4). This spectral clustering operation, using the normalized cut function in (4.4), assumes balanced cluster partitions to avoid trivial solutions such as finding a small cut that results in a single point belonging to its own cluster and everything else to another. Thus, the proposed ALPCAHUS method may not be ideal in situations with imbalanced clusters.

While one could use other clustering methods for imbalanced data, such as Ref. [15] or Ref. [133], these methods are not ideal in heterogeneous settings as they do not account for heteroscedasticity in the data. Instead, we recommend using the partition cut ("Pcut") method introduced in Ref. [134] that performs spectral clustering-like operations on a rank-modulated degree graph as a replacement for spectral clustering, which uses the normalized cut function ("Ncut") in (4.4). This would allow ALPCAHUS to be more robust in situations with imbalanced cluster sizes. Furthermore, the non-ensemble version of ALPCAHUS in (4.19) may also suffer in the imbalanced-cluster regime when the TIPS-based initialization scheme in Sec. 4.4.4 is used. This is due to the spectral clustering step in TIPS, which initializes cluster labels for ALPCAHUS. Therefore, even when $B = 1$, it is also recommended to apply the "Pcut" method in Ref. [134] on the affinity matrix formed by (4.39).

4.7.3 ALPCAHUS Parameters

For Alg. ALPCAHUS, the input data matrix $Y \in \mathbb{R}^{D\times N}$ and the number of subspaces or clusters $K \in \mathbb{Z}^+$ are necessary inputs that are clearly data dependent. For the candidate subspace dimension $\hat{d}_k$, it may be the same across all clusters or vary, and it is either known via domain knowledge or can be reasonably estimated by the procedure outlined in Sec. 4.4.3. If it must be estimated, it is recommended to start with an over-parameterized model rather than an under-parameterized one. A scree plot of Y can give an indication of sufficient rank for initialization to

ensure an over-parameterized state. Further, $\hat{d}_k$ may also be treated simply as a hyperparameter for cross-validation purposes.

For the threshold parameter $q \in \mathbb{Z}^+$, this can be set to the subspace dimension as shown in Ref. [122] and explained in Sec. 4.4.1, or simply treated as a hyperparameter to be optimized during cross-validation; this value is dataset dependent. The number of base clusterings $B \in \mathbb{Z}^+$ is either fixed to $B = 1$ for the non-ensemble version of ALPCAHUS, or set as high as compute time and memory allow for the ensemble version. In other words, more trials make it easier to identify the underlying relationships between similar data samples. If $B = 1$, then it is recommended to use the TIPS-based initialization scheme from Sec. 4.4.4.

Since each subspace basis update involves LR-ALPCAH [13], the $T_1 \in \mathbb{Z}^+$ parameter specifies the number of LR-ALPCAH iterations. For T_1, it is recommended to keep it low; e.g., we set $T_1 = 5$ since there is no point in converging perfectly, as the cluster labels will be updated in the next reassignment at $t_2 + 1$. For the non-ensemble version, it is recommended to do as many alternating updates $T_2 \in \mathbb{Z}^+$ as necessary to trigger the stopping criteria in line (10) of Alg. ALPCAHUS. However, for the ensemble version of ALPCAHUS, we find that a low value, such as $T_2 = 3$, works very well in practice due to the larger number of trials, $B \gg 1$. This value agrees with Fig. 4.9 as ALPCAHUS converges in just a few iterations anyway. For more detailed and practical code-oriented function usage, refer to the documentation at github.com/javiersc1/ALPCAHUS.

CHAPTER 5

PET-TURTLE: Deep Unsupervised Support Vector Machines for Imbalanced Data

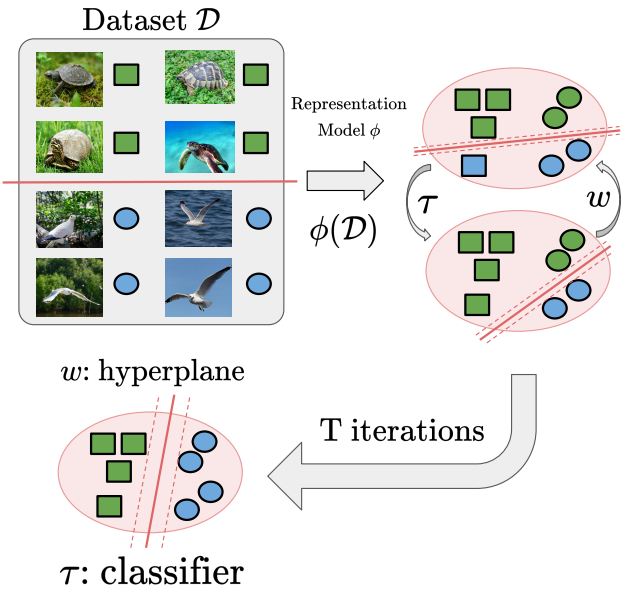


Figure 5.1: A visual illustration of the key idea behind unsupervised support vector machines that alternates updates between labels and hyperplane estimation.

5.1 Introduction

Transfer learning leverages pretrained neural networks to improve model performance on downstream tasks, often with limited data [135]. Recent works show that fine-tuning an entire model has only marginal improvements compared to using a frozen backbone with a linear classifier [136] [137] [138]. These self-supervised foundational models, trained in the CLIP or contrastive learning

This work in this chapter is published in IEEE Signal Processing Letters [15].

paradigm [139] [140], offer competitive performance on a variety of downstream tasks by learning general representations and using them in zero-shot settings. However, in some scenarios, data labels are unavailable, making it impossible to train a linear classifier on top of these models. Thus, one would naturally think about applying clustering methods such as K-Means [20] or subspace variations [14] on the latent representations. This strategy leads to worse performance compared to weakly supervised approaches [141].

In recent work, the TURTLE algorithm [142] was proposed to overcome such challenges by searching for a labeling that maximizes the hyperplane margin to uncover data groups in an unsupervised manner. Compared to other deep clustering methods such as DEC [143], DAC [144], DeepCluster [145], and SPICE [146], TURTLE does not use task-specific representation learning that is typically very expensive for modern foundation models. Compared to these methods, TURTLE achieved the new state-of-the-art performance in unsupervised clustering.

Diving further into the problem statement, let ϕ represent some backbone model such that $\phi(x) = z \in \mathbb{R}^d$ represents latent features belonging to d-dimensional space and associated with original data sample x from dataset $\mathcal{D}$. Let C correspond to the number of clusters. Let $\tau_\theta(z) : \{\mathcal{Z} = \phi(\mathcal{D})\} \to \{1, \ldots, C\}$ denote the classifier with continuous parameters θ that uncover the underlying labeling of $z \in \mathcal{Z}$. Specifically, let $\tau_\theta(z) = A_\theta z + b_\theta$ where $A_\theta \in \mathbb{R}^{C \times d}$ and $b_\theta \in \mathbb{R}^C$. Let $w_\theta \in \mathbb{R}^{C \times d}$ correspond to the hyperplane in the latent space that also includes a bias term. Then, the TURTLE optimization objective, given a single representation space, solves

$$\begin{aligned} \mathcal{L}_{\text{TURTLE}}(\theta) = &\sum_{z \in \phi(\mathcal{D})} \mathcal{L}_{\text{CE}}(w_\theta^M z; \sigma(\tau_\theta(z))) \\ &\text{s.t. } w_\theta^M = \Xi^{(M)}(w_\theta^M, \phi(\mathcal{D})) \end{aligned} \tag{5.1}$$

where $\sigma(\cdot)$ denotes the softmax operation and $\mathcal{L}_{\text{CE}}(\cdot)$ is the cross entropy loss [147]. The inner term $\Xi^{(M)}(w_\theta^M, \phi(\mathcal{D}))$ denotes an iterative optimization algorithm Ξ, such as gradient descent, run for M steps starting from a randomly initialized w_θ^0. In a binary classification setting with linearly separable data, (5.1) corresponds to unregularized logistic regression. In Ref. [148], the authors show that gradient descent applied to (5.1) with known labels induces iterates that are biased towards the direction of the maximum hard-margin hyperplane. Later work showed that similar results hold true in the non-separable setting for a soft-margin hyperplane [149]. TURTLE exploits a bi-level optimization problem by alternating between finding a hyperplane and using it to update labels, similar to K-Means but in a support vector machine context.

However, regularization is needed for the classifier τ to prevent the global and trivial solution where the encoder classifies all samples to the same cluster and thus technically achieves a minimum value for (5.1). In this $C = 1$ context, one can easily find maximum margin since the

hyperplane can be far away from the single cluster data corresponding to a low cost function value, as shown in Ref. [150]. Let

$$\bar{\tau}_\theta = \frac{1}{|\mathcal{D}|} \sum_{z \in \phi(\mathcal{D})} \tau_\theta(z) \tag{5.2}$$

be the empirical label distribution of $\mathcal{D}$ predicted by the classifier. Then, the final objective function of the TURTLE formulation is

$$\min_\theta \mathcal{L}_{\text{TURTLE}}(\theta) - \gamma \mathbb{H}(\bar{\tau}_\theta) \tag{5.3}$$

where $\mathbb{H}(\cdot)$ corresponds to the entropy function of discrete distributions. Let $h_i \in \bar{\tau}_\theta$ correspond to the ith element of the vector $\bar{\tau}_\theta$. Then, $\mathbb{H}(\cdot)$ operates element-wise by

$$\mathbb{H}(h_i) = \begin{cases} -h_i \ln h_i & h_i > 0 \\ 0 & h_i = 0 \\ -\infty & h_i < 0 \end{cases} \tag{5.4}$$

and this prevents the degenerate solution $C = 1$. However, it comes at the cost of encouraging balanced clusters since there is an equal penalty for all classes. Thus, the TURTLE formulation has an explicit, balanced cluster assumption that leads to worse clustering quality in the imbalanced data regime. This phenomenon is observed in self-supervised learning in general [151]. Some ways others have approached this problem, to give a few examples, are by ensemble methods that fuse information from different experts [152] or by randomly encoding subtasks to find the adaptive weighting during training [153].

In this chapter, we address the balanced cluster assumption by generalizing (5.3) to better handle distributions in imbalanced datasets, thereby improving accuracy and cluster cohesion. Our new method, named PET-TURTLE, achieves higher performance in the imbalanced regime depending on the severity of data imbalance. Additionally, our method even reduces clustering error in balanced data regimes because the sparse logit model of the classifier τ limits the number of potential classes to consider for hyperplane updates to only those with higher probabilities. We conjecture that this enables a more efficient hyperplane search that is not constrained by low-probability predictions. This is similar in scope to subspace learning problems where very noisy data samples can lead to a worse subspace basis estimate in principal component analysis (PCA) [12]. In Sec. 5.2, the proposed PET-TURTLE algorithm is discussed in greater detail, beginning with the cluster imbalance problem.

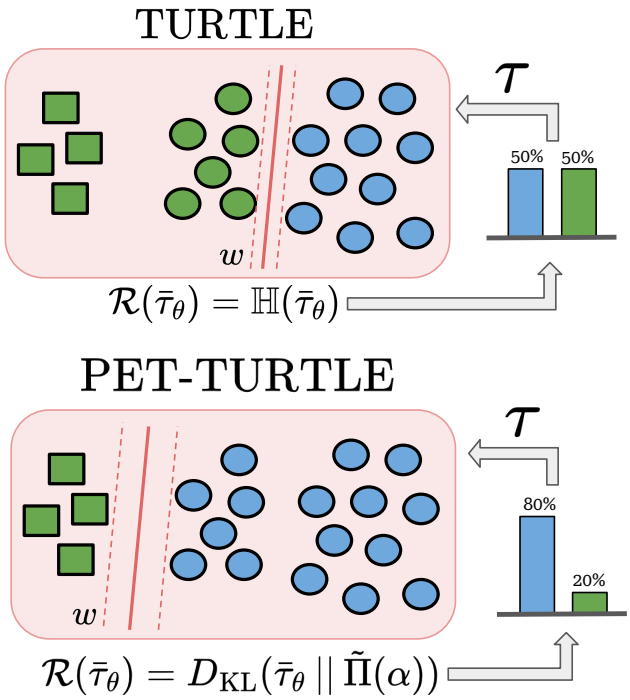


Figure 5.2: A visual comparison of the effects of regularization terms in the TURTLE and PET-TURTLE objective functions.

5.2 Proposed Method

5.2.1 Prior Enforcement Term for Imbalanced Data

To address the imbalanced cluster limitation, we consider the following class of optimization problems

$$\min_{\theta} \mathcal{L}_{\text{TURTLE}}(\theta) + \gamma D_{\text{KL}}[\bar{\tau}_\theta \parallel \mathbf{\Pi}] \tag{5.5}$$

where $D_{\text{KL}}(\cdot)$ corresponds to the KL-divergence [154] of the two distributions $\bar{\tau}_\theta$ and a prior data distribution $\mathbf{\Pi}$. In the balanced cluster regime, $\mathbf{\Pi}$ can be assumed to be uniform. In the imbalanced regime, $\mathbf{\Pi}$ can be known for some applications, e.g., in scene graph generation problems, where a KL prior is added to the cross-entropy loss in a classification context [155]. In many instances, the distribution is unknown, yet it can be assumed to be imbalanced.

We propose using a power-law distribution [156] to model imbalanced data as a proxy when the underlying distribution is unknown. This is a natural choice for imbalanced distributions due to the wide assortment of applications in physics, biology, medicine, chemistry, astronomy, and economics [156]. The power law probability mass function, given decay factor $\alpha \in \mathbb{R}_{\geq 0}$, is defined as

$$p_{\text{powerlaw}}(c; \alpha) = \frac{c^{-\alpha}}{\sum_{x=1}^{C+1} c^{-\alpha}} \tag{5.6}$$

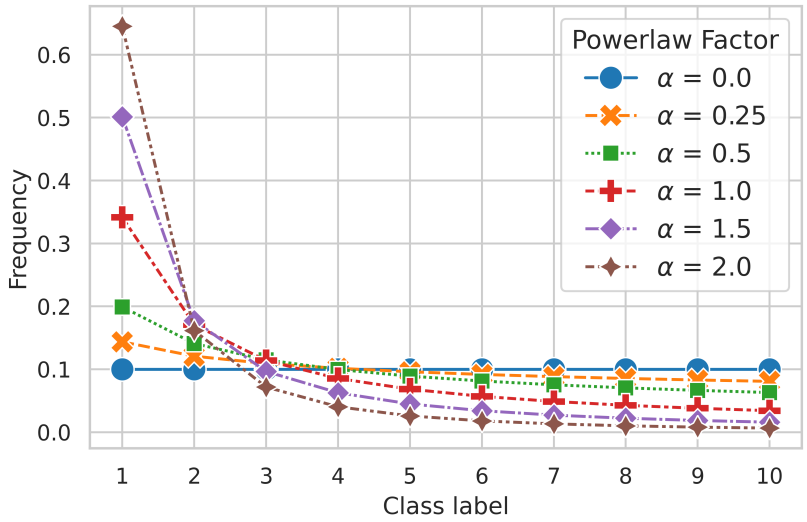


Figure 5.3: Probability mass functions of the power law distribution at various decay rates α when $C = 10$ as used in Table 5.1.

where $c \in \{1, \dots, C\}$. Let $\tilde{\mathbf{\Pi}}(\alpha) \in \mathbb{R}^C_{\geq 0}$ represent the vectorized mass function for the power law distribution such that $\forall c,\ p_{\text{powerlaw}}(c; \alpha) \in \tilde{\mathbf{\Pi}}(\alpha)$. See Fig. 5.3 for a plot illustrating the power law distribution when $C = 10$ at different α values.

Since the distribution has a statistical parameter α that describes data imbalance, one could use domain knowledge or known labels to select a reasonable value. For applications where the imbalance is not known, α can be estimated. One can find optimal (γ, α) values by using estimated labels $\{y_1, \dots, y_N\}$ from $y_i = \arg\max \tau_\theta(\phi(x_i))\ \ \forall i \in \mathcal{D}$ and measure the generalization error of the linear classifier trained on those estimated labels in a cross validation setting as similarly done for γ in Ref. [142]. This process does not require ground-truth labels; instead, it measures the hyperplane margin based on the algorithm-derived labels. Because of this, no prior knowledge is necessary besides suspecting imbalanced data in a given problem setting.

5.2.2 Sparse Logits for Hyperplane Estimation

From (5.1), observe that $\mathcal{L}_{\text{CE}}(w_\theta^M z; \sigma(\tau_\theta(z)))$ is using soft labels to find the optimal hyperplane. This is because a discrete search space contains $\mathcal{O}(C^N)$ possible labelings, which is an NP-hard problem [157]. The search space is restricted with continuous parameters θ in $\tau_\theta(x) = A_\theta x + b_\theta$ to enable efficient gradient optimization. However, due to the softmax operation $\sigma(\tau_\theta(z))$, every logit plays a role in updating the hyperplane, even low-value logits that are unlikely to be the correct label for a fixed data point. This is due to the full support of the softmax function, and we conjecture from experimentation that this may cause a non-ideal estimation of the hyperplane, given the results in Table 5.2.

Instead, we propose to filter low-value logits by applying the sparsemax function [158] defined as solving

$$\text{sparsemax}(z) = \underset{p \in \Delta^{C-1}}{\arg\min} \|p - z\|_2^2 \tag{5.7}$$

where Δ^{C-1} corresponds to the probability simplex. This operation returns the Euclidean projection of z onto the simplex. Simplex projection is a problem with efficient solutions that involve soft thresholding smaller values with automatically chosen thresholds [159]. We use this idea in the loss function and propose a modified clustering loss with sparse simplex projection as

$$\begin{aligned} \mathcal{L}_{\text{SSP}}(\theta) &= \sum_{z \in \phi(\mathcal{D})} \mathcal{L}_{\text{CE}}(w_\theta^M z; \text{sparsemax}(\tau_\theta(z))) \\ &\text{s.t. } w_\theta^M = \Xi^{(M)}(w_\theta^M, \phi(\mathcal{D})) \end{aligned} \tag{5.8}$$

with the overall objective that includes the prior term being

$$\min_\theta \mathcal{L}_{\text{SSP}}(\theta) + \gamma D_{\text{KL}}[\bar{\tau}_\theta \parallel \tilde{\mathbf{\Pi}}(\alpha)]. \tag{5.9}$$

This variant is denoted as **PET-TURTLE** (**P**rior **E**nforcement **T**erm **TURTLE**). For completeness, algorithm pseudocode for PET-TURTLE is provided in Alg. 3. Additionally, Pytorch code is provided at github.com/javiersc1/pet-turtle for the method.

Algorithm 3 PET-TURTLE

Input: Dataset $\mathcal{D}$, representation model ϕ, classes C, and regularization parameters (γ, α)
Parameters: Iterations $T = 6000$, learning rate $\eta = 10^{-3}$, and inner steps $M = 10$
Extract latent variables $\mathcal{Z} = \phi(\mathcal{D})$
for $t = 1$ to T **do**
  Sample mini-batch representations $z \sim \mathcal{Z}$
  // update plane given fixed logits sparsemax$(\tau(z))$
  $w_\theta^M \leftarrow w_\theta^M - \eta \frac{\partial}{\partial w_\theta}$ [cost function in (5.9)] for $1, \ldots, M$
  // update classifier given fixed hyperplane ω^M
  $\tau_\theta \leftarrow \tau_\theta - \eta \frac{\partial}{\partial \tau_\theta}$ [cost function in (5.9)]
  if warm-start **then**
    // use latest plane to initialize for next iteration w_θ^0
    update start point $w_\theta^0 \leftarrow w_\theta^M$
end for
Output: Cluster labels $\arg\max \tau_\theta(\mathcal{Z})$

		K-means++[160]	TURTLE[142]	PET-TURTLE	Linear Probe[161]
CIFAR10-PL	Powerlaw($\alpha = 0.25$)	65.5 ± 4.0	72.8 ± 0.3	**78.7*** ± 2.6	93.8
	Powerlaw($\alpha = 0.50$)	60.3 ± 4.3	68.2 ± 1.3	**74.0*** ± 2.0	93.8
	Powerlaw($\alpha = 1.0$)	50.9 ± 2.8	54.9 ± 3.0	**71.5*** ± 3.5	94.6
	Powerlaw($\alpha = 1.50$)	42.5 ± 3.0	48.3 ± 4.8	**67.1*** ± 4.1	95.0
	Powerlaw($\alpha = 2.0$)	32.8 ± 2.6	42.8 ± 5.0	**60.6*** ± 3.9	96.2

Table 5.1: Accuracy results (%) of clustering methods on the CIFAR10-PL dataset at power-law decay rates.

5.3 Experiments & Results

5.3.1 Experimental Setup

For all of our experiments, CLIP-RN50x64 [139] is used for feature extraction, meaning only a single representation space is used. We compare PET-TURTLE against TURTLE [142] and include the K-Means++ algorithm [160] to establish a baseline. Additionally, linear probing [161], the *supervised* linear classifier method, is included to determine the highest accuracy possible assuming the labels are known for the associated latent features. It is worth noting that, in general, clustering is a harder problem than classification so results can vary greatly depending on the difficulty of the problem. For all results, the average and standard deviation across 10 trials are reported, with different model seed initializations where applicable, for transparency. Furthermore, paired t-tests were conducted between TURTLE and PET-TURTLE trials, and results with p-values less than 0.01 are indicated with the "*" symbol.

5.3.2 Synthetic Results

We generated a power law imbalanced dataset based on CIFAR10 [162] at various decay rates $\alpha \in \{0.25, 0.50, 1.0, 1.50, 2.0\}$ and denote this variant as "CIFAR10-PL". Since TURTLE and PET-TURTLE have a regularization parameter γ, cross-validation is used to find the one that leads to the lowest generalization error in the set of validation values $\gamma \in \{1, 5, 10, 25, 50, 100, 250, 500\}$. Because the decay rate α is known in this synthetic experiment, it is fixed for PET-TURTLE. Table 5.1 compares PET-TURTLE against the other methods at different α decay rates. From this result,

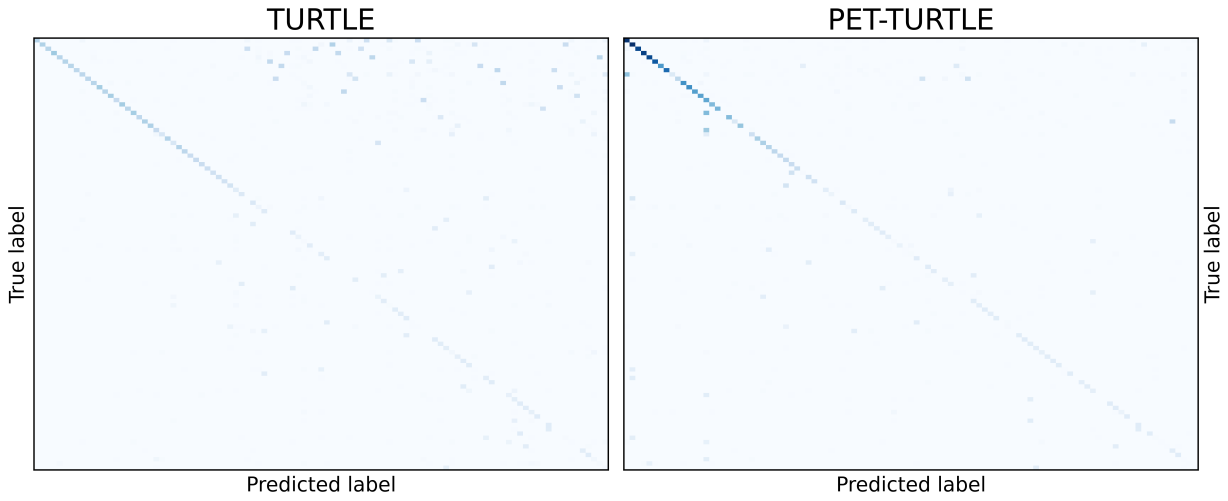


Figure 5.4: Confusion matrices of the TURTLE and PET-TURTLE methods on the Food101-PL dataset ($C = 101$) with fixed decay rate $\alpha = 1.0$.

it is clear that PET-TURTLE achieves higher clustering accuracy, especially at higher levels of data imbalance.

Similarly, a synthetic power-law dataset based on Food101 [163] is generated at a fixed $\alpha = 1.0$, contains $C = 101$ classes, and is denoted as "Food101-PL". In this experiment, only TURTLE and PET-TURTLE are compared to study the effects of over-prediction. Confusion matrices between the ground truth labels and the predicted cluster labels are generated in Fig. 5.4. The "Hungarian method" [123] is used to solve the cluster label permutation problem to ensure optimal correspondence between the true labels and the cluster labels. In Fig. 5.4, the TURTLE confusion matrix contains many elements in the upper triangular region, indicating that many samples that belong in the majority clusters were clustered into the minority clusters, thus leading to worse clustering. This is in contrast to PET-TURTLE, which not only exhibits fewer off-diagonal elements but also shows a more accurate color shift from dark blue to light blue in the counts for each cluster, as expected in an imbalanced data problem.

5.3.3 Real Data Results

Lastly, real data comparisons are made on balanced and imbalanced datasets in Table 5.2. For the balanced datasets, the following are used: Caltech101 [164], CIFAR10 [162], DTD [165], EuroSAT [166], and Food101 [163]. For the imbalanced datasets, the following are used: blood cell microscope (Blood) [167], dermatoscope (Derma) [168], iNaturalist 2017 [169], retinal OCT (OCT) [170], axial abdominal CT (OrganA) [171], and kidney cortex microscope (Tissue) [172]. The MedMNIST [173] package is used to load medical imaging datasets. The number of classes

	Dataset	C (num. of classes)	K-means++ [160]	TURTLE [142]	PET-TURTLE	Linear Probe [161]
Balanced	Caltech [164]	101	76.6 ± 4.5	85.3 ± 0.9	**88.2*** ± 0.5	96.9
	CIFAR [162]	10	61.7 ± 4.2	76.3 ± 0.1	**80.9*** ± 1.6	94.1
	DTD [165]	47	48.2 ± 1.8	57.6 ± 0.9	**59.4** ± 1.5	82.5
	EuroSAT [166]	10	60.0 ± 6.4	77.6 ± 0.2	**80.4*** ± 1.6	95.2
	Food [163]	101	63.1 ± 1.6	85.4 ± 1.1	**88.1*** ± 1.0	94.4
Imbalanced	Blood [167]	8	44.9 ± 0.7	48.1 ± 1.8	**56.1*** ± 2.7	96.1
	Derma [168]	7	25.4 ± 0.7	34.1 ± 0.3	**67.1*** ± 0.3	81.8
	iNaturalist [169]	13	43.6 ± 1.7	55.9 ± 3.7	**77.5*** ± 1.0	98.1
	OCT [170]	4	46.2 ± 0.9	54.1 ± 0.7	**61.4*** ± 0.7	93.4
	OrganA [171]	11	43.5 ± 2.8	45.0 ± 1.1	**51.9*** ± 2.4	89.9
	Tissue [172]	8	26.3 ± 0.2	30.6 ± 1.8	**38.7*** ± 1.2	60.5

Table 5.2: Accuracy results (%) of clustering methods on real balanced and imbalanced image datasets.

C is known for all datasets. Cross-validation is done similarly to before. However, in this instance, α is unknown and must be estimated. Grid search is used on the data imbalance parameter

$$\alpha \in \{0.01, 0.05, 0.10, 0.25, 0.50, 0.75, 1.0, 1.25, 1.50, 1.75, 2.0\} \tag{5.10}$$

and the pair (γ, α) that achieves the lowest validation error is selected. On average, in the balanced data regime, PET-TURTLE achieves a $\sim$3% accuracy improvement over TURTLE, thanks to the sparse-logit idea used for hyperplane estimation. In the imbalanced regime, on average, PET-TURTLE achieves a $\sim$15% improvement in accuracy over TURTLE, thanks to the prior enforcement term. It is important to note that the imbalanced datasets do not strictly follow a power-law distribution, so further improvements in clustering quality may be possible. In a way, this tests the robustness of our method against distribution mismatch.

5.4 Conclusion

In summary, this chapter presents a principled extension to the TURTLE clustering algorithm that addresses its limitations in imbalanced data settings by introducing a prior distribution term. This

approach demonstrates robust improvements in clustering accuracy across both synthetic and real-world scenarios, particularly in settings with significant class imbalance. These empirical gains highlight PET-TURTLE's practical relevance for unsupervised clustering applications atop foundation models, paving the way for more reliable clustering methods in the presence of imbalanced data distributions.

However, this proposed method relies on features extracted from a foundation model trained in some self-supervised paradigm. Thus, if one applies TURTLE or PET-TURTLE directly to ambient space data ($x \in \mathcal{D}$) that exhibits nonlinear structure, then neither method would work well since they expect a more linearly separable space, such as one induced by foundation models. On a related note, this is why *kernel* SVMs are more powerful than traditional SVMs for nonlinear settings. It would be interesting to explore an extension to PET-TURTLE in this nonlinear regime. This could be useful in a foundation model context, since there is likely some nonlinearity in the latent space ($z \in \phi(\mathcal{D})$), depending on the problem's difficulty or the training distribution, that makes it more challenging to find the optimal hyperplane. Further exploration on this topic would be an interesting direction for future work.

5.5 Impact Statement

The main goal of this chapter is to advance the field of clustering, and the proposed method relies on the representation spaces of foundation models. It is known that these models inherit biases embedded in the training data [174]. Because of this, caution is recommended when using PET-TURTLE in sensitive areas such as medical imaging.

CHAPTER 6

Alzheimer's Disease Diagnosis in Functional MRI via 4D Convolutions

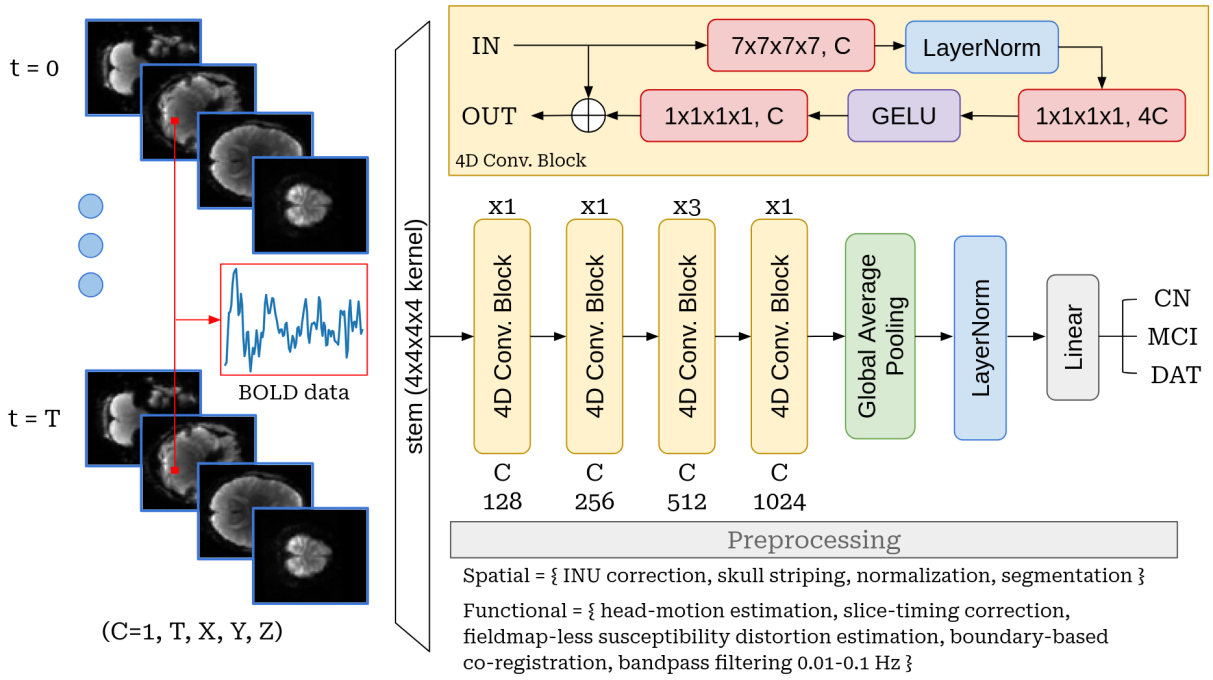


Figure 6.1: Proposed architecture, consisting of four downsampling stages in a 1-1-3-1 configuration. The final stage outputs 1024 channels, which are globally averaged to yield 1024 features.

6.1 Introduction

Resting-state functional MRI (rs-fMRI) is increasingly recognized as a biomarker for AD, with numerous studies reporting different blood-oxygen-level-dependent (BOLD) activations in specific

This chapter appeared as an abstract in the International Society for Magnetic Resonance in Medicine (ISMRM) conference during the 2025 annual meeting [16].

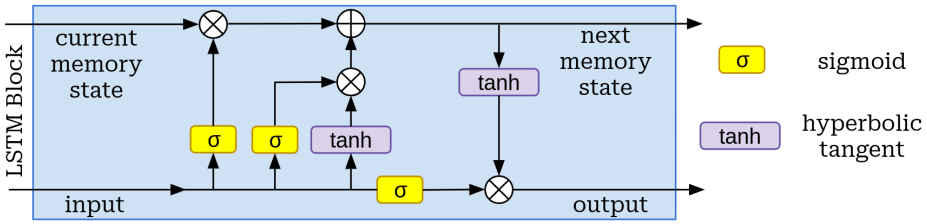


Figure 6.2: Illustration of an LSTM layer used in deep learning models.

brain regions relative to healthy subjects [175] [130]. Feature extraction from this neuroimaging data typically involves machine learning algorithms applied to either functional connectivity matrices or subcortical surface maps [175] [176]. Conversely, this study focuses on 4-dimensional data for classification, which is often overlooked due to greater computational demands. Since other approaches, such as timeseries data or functional connectivity matrices, reduce data dimensionality, some important information may be lost that is beneficial for diagnosis.

We evaluate three deep learning approaches for handling 4D data. The first approach employs a 3D convolutional neural network (CNN) using the ConvNeXt [177] architecture, treating time samples as input channels. The second approach is a hybrid model combining a 3D CNN with a long short-term memory (LSTM) [178] module to separately capture spatial features and temporal dynamics. The third approach, our method, introduces a novel 4D CNN model that performs convolutions using 4D temporal-spatial kernels. While using a 4D CNN is not entirely unprecedented [179] [180], this study represents the first application of such a model in fMRI for diagnosing Alzheimer's disease.

6.2 Background

6.2.1 Long Short-Term Memory (LSTM)

A Long Short-Term Memory (LSTM) [178] is a type of recurrent neural network (RNN) architecture that is particularly well-suited for modeling sequential data and capturing long-range dependencies. The LSTM addresses the vanishing gradient problem, which can arise during the training of traditional RNNs. It does this by introducing a more complex, gated architecture. LSTMs are powerful because they can maintain information across long sequences using the cell state and multiple gates to control information flow. As a result, they are commonly employed in areas such as language modeling, time-series prediction, and other applications where capturing temporal dependencies is critical. The architecture for an LSTM layer is described below.

An LSTM cell comprises several components designed to regulate the flow of information, namely the cell state C_t, the hidden state h_t, and the gates f_t, i_t, and o_t. The cell state C_t is

the LSTM cell's internal memory that carries information across time steps. It is modulated by various gates to retain essential information over long periods. The hidden state h_t is the output of the LSTM cell at time step t, which is used both as output and as input to other model components at the next time step. The LSTM cell contains three key gates. The first one, forget gate f_t, determines which information from the cell state should be discarded. It uses a sigmoid activation function to produce values between 0 and 1, which are then applied element-wise to the cell state, written as

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f). \tag{6.1}$$

The input gate i_t decides which new information should be added to the cell state. It also uses a sigmoid function to serve as a filter for input modulation, as expressed below

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i). \tag{6.2}$$

The output gate o_t determines which part of the cell state to output as the hidden state at the current time step. This gate uses sigmoid functions as a gate mechanism before computing the new hidden state, as indicated

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o). \tag{6.3}$$

The cell state is updated using the forget gate and the candidate updates. This layer generates potential values to add to the cell state, typically using a hyperbolic tangent activation function. The new cell state is computed along with the hidden state using the output gate and cell state. Illustrated in Fig. 6.2, this is mathematically written as

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{6.4}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{6.5}$$

$$h_t = o_t \odot \tanh(C_t). \tag{6.6}$$

6.2.2 Global Average Pooling (GAP)

Global Average Pooling (GAP) [181] is a data processing technique that distills a high-dimensional data representation into a simplified form by computing the average across particular dimensions of a tensor, in this context used to both temporally and spatially compress the tensor. GAP operates by aggregating information across the entire specified dimensions, thereby condensing a feature set into a single representation for each channel or feature dimension. This operation effectively transforms a multidimensional data array into a lower-dimensional tensor, often reducing the dimensions associated with spatial characteristics to a single mean value per feature channel. Since

our data is size $(B, C, S_X, S_Y, S_Z, S_T)$ for B batch size, C channels after many convolutional layers with reduced spatial and time dimensions (S_X, S_Y, S_Z, S_T), GAP will reduce the input to size (B, C) by averaging over (S_X, S_Y, S_Z, S_T) in this context. Note that (S_X, S_Y, S_Z, S_T) is very small in general at the end of any CNN, in our case $3 \times 3 \times 3 \times 4$. This process yields a compact representation, facilitating the use of a linear layer for classification. Unlike many other neural network operations that require parameter learning, GAP functions independently of learned parameters. GAP can handle input data of variable sizes because the averaging process is independent of the absolute dimensions, focusing instead on the mean value across dimensions. This property is particularly advantageous in this situation, as the original input size may vary depending on the spatial voxel size and the repetition time (sampling rate).

6.2.3 Layer Normalization

Layer Normalization [182] is a technique used in neural networks to improve training stability and convergence by normalizing the activations of intermediate layers. It addresses the problem of internal covariate shift, where the distribution of inputs to a given layer changes during training, potentially slowing learning. Consider a layer with activations represented by a vector $\mathbf{x} = [x_1, x_2, \ldots, x_H]$, where H denotes the number of hidden units in the layer. For layer normalization, the mean and variance are computed across the layer's units

$$\mu = \frac{1}{H} \sum_{i=1}^{H} x_i \tag{6.7}$$

$$\sigma^2 = \frac{1}{H} \sum_{i=1}^{H} (x_i - \mu)^2. \tag{6.8}$$

Then, each element x_i is then normalized using the computed mean and variance:

$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{6.9}$$

where ϵ is a small constant added for numerical stability to prevent division by zero. Finally, just like batch normalization, the normalized activations are scaled and shifted using learnable parameters γ (scale) and β (shift) by

$$y_i = \gamma \hat{x}_i + \beta. \tag{6.10}$$

Here, γ and β are learned parameters that allow the network to adapt the normalized output to the desired range of activations.

Layer normalization normalizes each data point independently across its features, unlike batch normalization, which normalizes across the batch dimension. This characteristic makes layer normalization particularly useful for recurrent neural networks or scenarios with small batch sizes. By stabilizing the distribution of inputs to each layer throughout training, layer normalization can accelerate convergence and potentially improve performance. The additional computational complexity of layer normalization is relatively low, as it involves only computing per-layer statistics and learning a linear transformation.

6.2.4 Gaussian Error Linear Unit (GELU)

The Gaussian Error Linear Unit (GELU) [183] is an activation function used in neural networks. It differs from traditional activation functions such as ReLU (Rectified Linear Unit) by incorporating stochastic regularization, which has been shown to improve performance across several tasks. The GELU activation function is defined as

$$\text{GELU}(x) = x \cdot \Phi(x) \tag{6.11}$$

where $\Phi(x)$ is the cumulative distribution function (CDF) of the standard normal distribution. This can be approximated using either $\text{erf}(\cdot)$, the Gaussian error function, or using hyperbolic tangent as follows

$$\Phi(x) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right] \tag{6.12}$$

$$\Phi(x) \approx \frac{1}{2} \cdot \left(1 + \tanh\left(\sqrt{\frac{2}{\pi}}\left(x + 0.044715 \cdot x^3\right)\right)\right). \tag{6.13}$$

Unlike deterministic activations such as ReLU, GELU introduces stochasticity by weighting inputs based on how they compare to their own normal distributions. This allows GELU to decide which neurons to activate more gradually than ReLU's hard thresholding. Because of this, GELU behaves smoothly across the input range by preventing zero gradients, a known problem with ReLU on negative inputs. By employing a Gaussian-based nonlinear activation function, GELU enables networks to process input data more smoothly across its activation spectrum, allowing them to learn more complex decision boundaries and general nonlinear relationships in the data.

6.2.5 4D Kernels

Four-dimensional spatial-temporal convolutions are aimed at processing data that possesses both spatial and temporal dimensions. Such operations are beneficial when the data evolves over time,

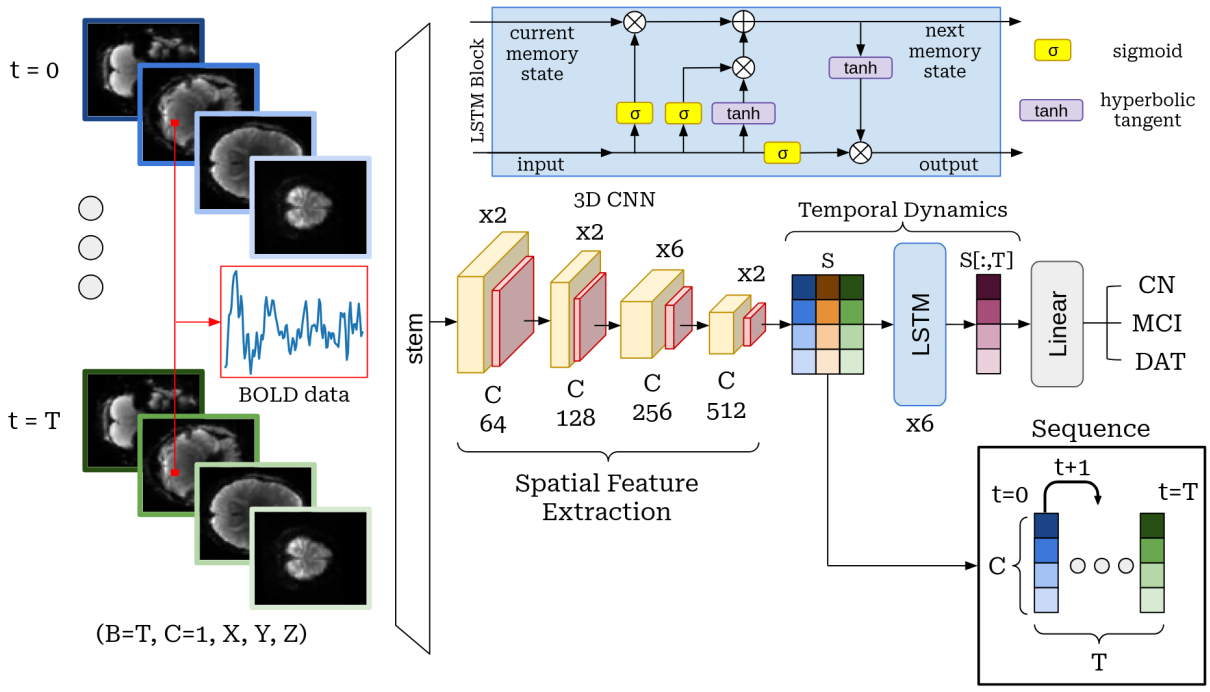


Figure 6.3: Architecture of the hybrid 3D CNN + LSTM model. Each 3D time sample is processed individually by the CNN, and the resulting features are aggregated into the matrix S. The LSTM module captures temporal dynamics between time samples for classification.

necessitating a model capable of capturing dependencies across both spatial and temporal dimensions. Incorporating 4D convolutions extends traditional convolution operations to handle datasets that are represented in a $C \times X \times Y \times Z \times T$ format. Here, C denotes the number of input channels, while X, Y, and Z correspond to spatial dimensions, and T signifies the temporal dimension.

In essence, a 4D convolution layer applies a series of learnable filters (or kernels) across the input data, capturing local patterns not only spatially but also across temporal sections. This operation simultaneously processes information across these dimensions, leading to a comprehensive understanding of how spatial features change, remain consistent, or evolve over time or across sequences. For instance, in video analysis, understanding the correlation and progression of scenes is essential. Additionally, in medical imaging, 4D convolutions might be employed to analyze dynamic sequences of volumetric data, such as in functional MRI scans, where spatial patterns within bodily structures need to be interpreted over time to gain further insights. Thus, a 4D CNN is a robust tool for modeling and understanding datasets with inherent multidimensional relationships, significantly enhancing neural networks' capacity to handle temporally rich data structures.

6.3 Methods

6.3.1 Dataset & Processing

We used the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset [2], selecting 3T rs-fMRI scans (MPRAGE/SPGR pulse sequence data) with spatial resolution 3.3mm and a repetition time of 3000ms, comprising 140 temporal samples of size 65×77×65. Additionally, structural MRI scans were used to assist in the normalization process. Recognizing the usually limited size of medical datasets, we augmented the dataset by considering each session as an independent "pseudo-subject" to mitigate class imbalance and increase dataset size. To prevent cross-contamination between the train and test sets, scans from the same individual were assigned exclusively to one set, resulting in class distributions CN (602/50), MCI (210/50), and DAT (147/50) for the train/test sets. The test set was balanced to ensure that accuracy was a meaningful metric, and the validation set was a subset of the training set obtained via k-fold cross-validation. Data preprocessing involved several steps: converting raw DICOM files to the BIDS format [184] and processing with fmriprep [185]. For structural scans, this included N4 bias field correction, skull stripping, and spatial normalization to the MNI152 linear space from TemplateFlow [186]. For functional scans, this entailed slice-timing correction, head-motion estimation, and fieldmap-less susceptibility distortion correction. Further preprocessing involved bandpass filtering between 0.01-0.1 Hz using scikit-learn [187], discarding the first 20 temporal samples, and applying Z-score normalization on each voxel's time series data.

6.3.2 4D CNN Model

All models were implemented using PyTorch, with data importation facilitated by the NiBabel [188] package. We addressed class imbalance using a weighted cross-entropy loss function with inverse frequency weights $w = [\frac{959}{602}, \frac{959}{210}, \frac{959}{147}]$ and used the Adam optimizer [189] with weight decay regularization and a cosine decay learning rate scheduler. Training was conducted on a system equipped with a 32-core CPU, 187GB of RAM, and 1 NVIDIA 4090 GPU with 24 GB of VRAM. For the 4D CNN model, custom "Conv4D" layers were developed and integrated into our 4D convolutional blocks as illustrated in Fig. 6.1. Effectively, both time and space are considered in the convolution process, and the data is treated as having a single input channel. After the 4D backbone, average pooling is applied across the spatial and temporal dimensions, yielding a 1024-dimensional vector. This vector is provided to a linear layer for classification.

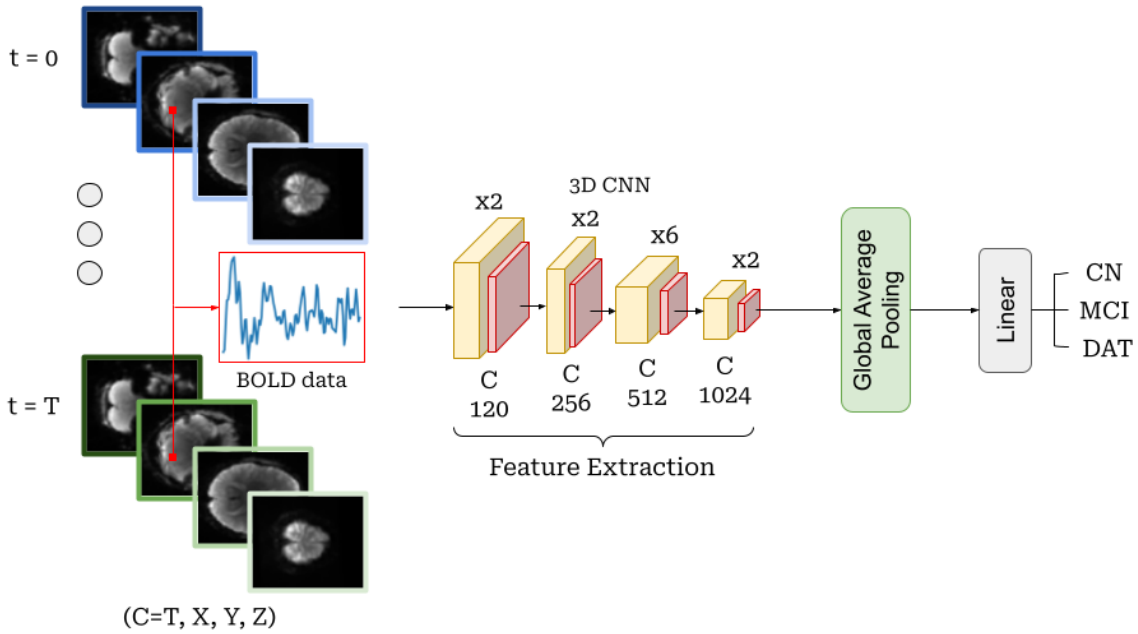


Figure 6.4: Architecture of the 3D CNN. This approach treats different time samples as channels in the conventional 3D CNN, with channel sizes increasing in the intermediate representations. The representation is average-pooled over spatial dimensions, and a linear layer is used for classification.

6.3.3 3D CNN + LSTM Model

For the 3D CNN + LSTM model, spatial features for each time sample were separately extracted and globally averaged. This is done by treating each time sample as part of the mini-batch, which means a channel size of 1. Then, all of the collected time samples are aggregated in temporal order into the matrix S. This sequence matrix is provided to the LSTM module, where the last temporal activation vector, $S[:, T]$, is used for classification by a linear layer, as shown in Fig. 6.3.

6.3.4 3D CNN

For the 3D CNN model, all time samples were treated as separate channel inputs, and the model goes from an initial 120 channels to 1024 channels by the end of the network. The spatial dimensions are averaged, and the resulting 1024-dimensional vector is the input to the linear layer for classification.

Method	Accuracy	Sensitivity	Specificity
2 class (CN/DAT)			
3D CNN	0.68	0.54	0.84
3D CNN + LSTM	0.72	0.58	0.88
4D CNN	0.77	0.62	0.92
2 class (CN/MCI)			
3D CNN	0.58	0.66	0.50
3D CNN + LSTM	0.61	0.40	0.82
4D CNN	0.63	0.64	0.62
3 class (CN/MCI/DAT)			
3D CNN	0.48	0.48	0.82
3D CNN + LSTM	0.50	0.50	0.78
4D CNN	0.53	0.53	0.83

Table 6.1: Comparative results for the three approaches to handling the time dimension in raw 4D fMRI data. Accuracy, sensitivity, and specificity are reported for various class settings (binary and multi-class classification) using the ADNI test dataset.

6.4 Results

6.4.1 Model Comparisons

In Table 6.1, various 2-class and 3-class settings are conducted with the three proposed model approaches, with reported accuracy, sensitivity, and specificity values. The 4D CNN model better predicted patient diagnoses than other models.

6.4.2 Model Interpretability

For model interpretability, two analyses were conducted. First, the 4D kernels in the first layer were plotted against time to visualize features learned by the model in Fig. 6.5. From this, it appears that some lower-level features include derivative and averaging kernels. Secondly, the Grad-CAM++ method [190] was employed to identify important regions used for diagnosis. Fig. 6.6 presents the BOLD response in the hippocampus with corresponding Grad-CAM signal, and to the right, spatial saliency maps at a fixed time point, highlighting significant spatial features such as cerebellum, prefrontal cortex, and hippocampus. One can observe that saliency changes for each time point for all regions, and different regions will have a different saliency response.

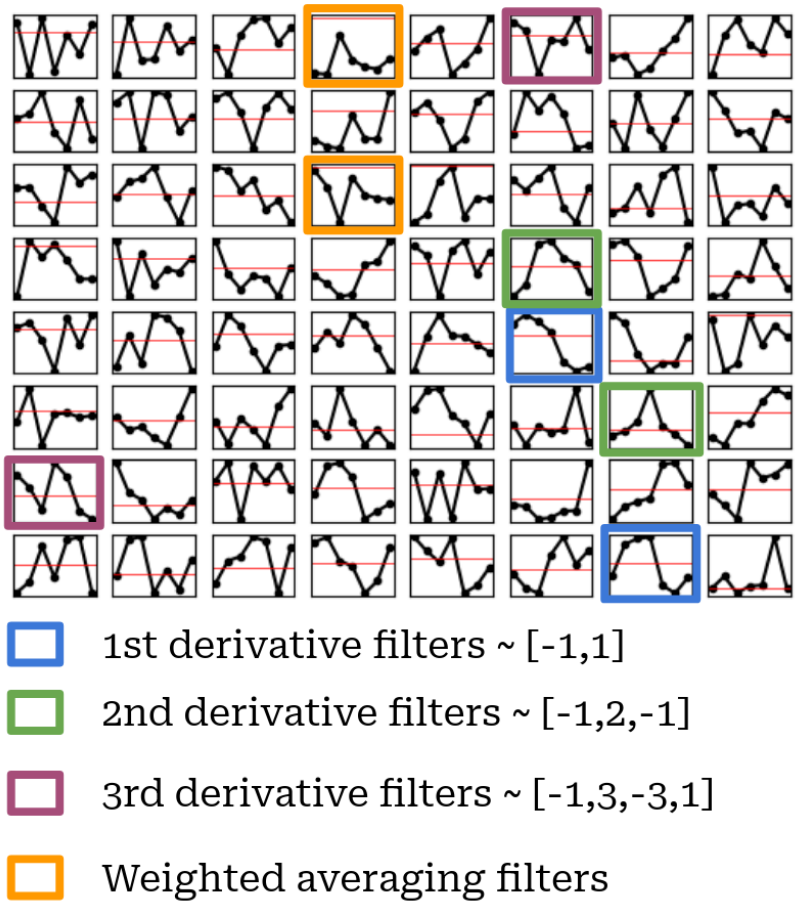


Figure 6.5: Temporal kernels from random spatial kernel locations for first layer channels (C=128). Only a subset of the total channels is shown for illustration simplicity. Moreover, only a few examples per filter are shown. The proposed model in the first layer extracts low-level features by using derivative and weighted average filters, among other kernels less interpretable.

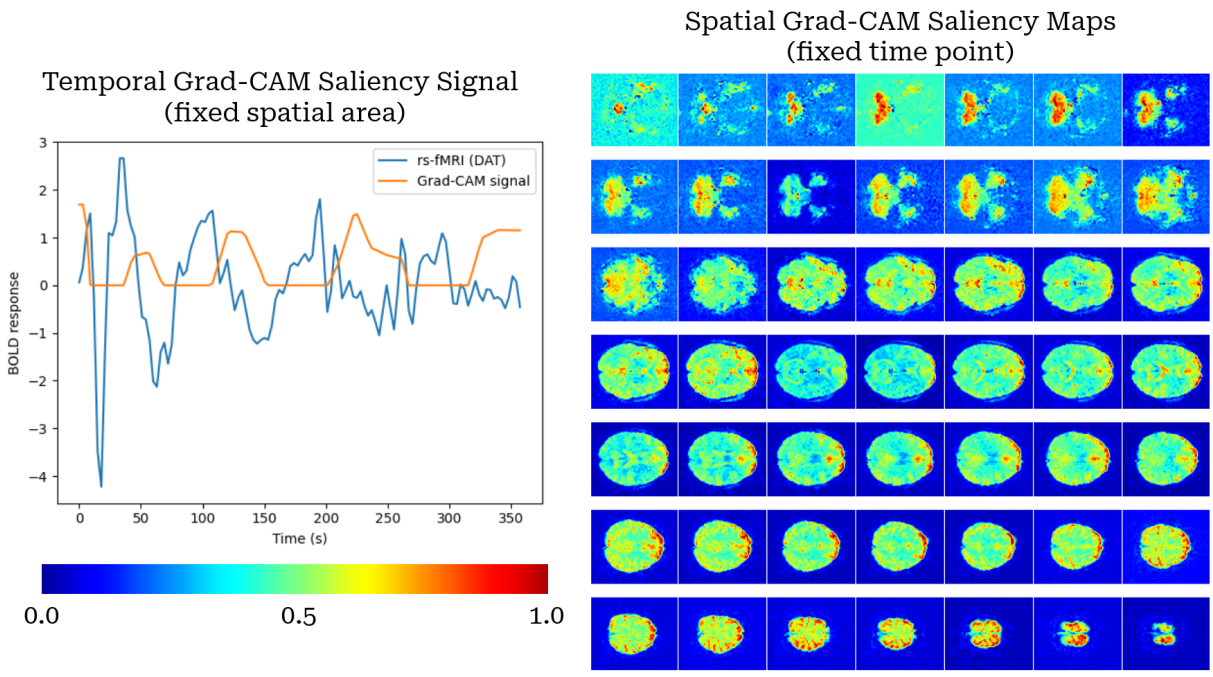


Figure 6.6: Model interpretability figure using the Grad-CAM++ method. The left image shows the BOLD response in the hippocampus for a DAT-diagnosed subject and the corresponding Grad-CAM saliency signal over time. The right image shows spatial Grad-CAM maps at a fixed time sample, illustrating the key regions used for classification.

6.5 Conclusion

This study demonstrates that diagnosis can be predicted using joint temporal-spatial kernels, illustrating the efficacy of a 4D CNN. The model outperformed other methods that rely on more conventional modeling assumptions, such as separate spatial and temporal learning modules. Moreover, saliency maps indicate relevant brain regions used for diagnosis.

Future research directions can focus on extending task-based fMRI data by targeting additional networks beyond the default mode network with stressors, to provide more insight into cognitive performance and Alzheimer's diagnosis. Additionally, the model could be restructured to work on regression-based tasks, such as score prediction, rather than classification.

CHAPTER 7

Behavior Score Prediction in Resting-State and Task-Based Functional MRI

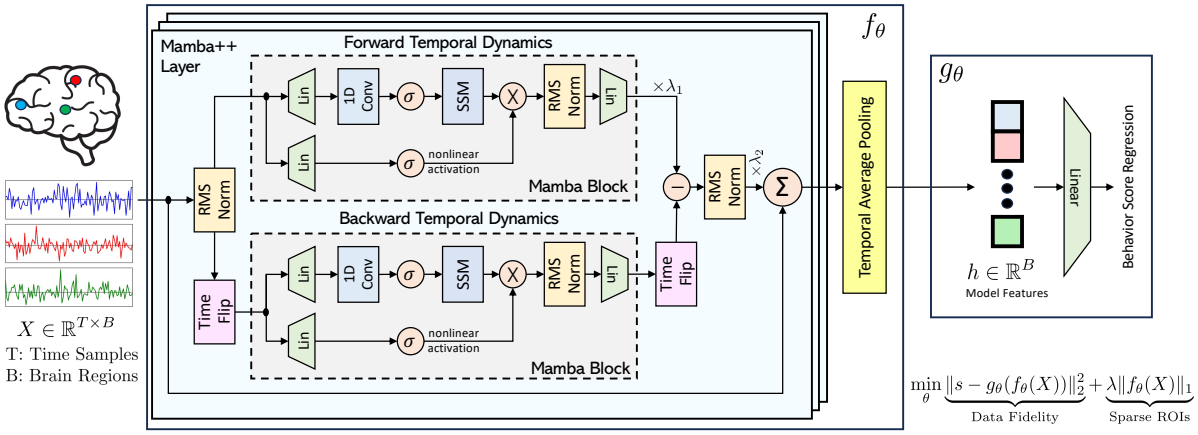


Figure 7.1: Overview of the proposed NeuroMamba architecture for behavior score prediction using deep state space modeling. The Mamba++ layer extracts temporal features from each brain region relevant to prediction. Temporal averaging is subsequently applied to derive a single scalar summary statistic per region, which is then processed by a linear head.

7.1 Introduction

7.1.1 Background

Functional magnetic resonance imaging (fMRI), which consists of a series of volumetric brain MRI scans, has been explored in various contexts to provide insights into brain function [191].

The work associated with the resting-state fMRI data will be submitted to IEEE Journal of Biomedical and Health Informatics [17]. Preliminary results on the task-based fMRI data are included in this chapter, which currently remain unpublished and require additional work before publication.

The signal acquired, namely blood-oxygen-level-dependent (BOLD) data, is used as a proxy for neural activity that impacts hemodynamics at the voxel level. Resting-state fMRI (rs-fMRI) is a neuroimaging technique that measures spontaneous brain activity by detecting regional differences in cerebral blood flow while a person is not performing any explicit task under the scanner. It is important because it reveals patterns of functional connectivity between brain regions, providing insights into brain organization and various neurological or psychiatric conditions [192]. These patterns indicate correlated brain activations that can reflect physiological changes in the brain [9, 193]. Recent work has focused on identifying neurological diseases using machine learning models based on functional connectivity [194, 195], a technique that measures correlations in BOLD activity among brain regions. Further, various studies have shown that the default mode network (DMN), a collection of brain regions active when a person is not performing a specific task in rs-fMRI, constitutes a baseline state that involves self-referential processing and internally directed cognition [7, 8]. In this chapter, we explore whether the default mode network plays a significant role in cognitive impairment by predicting behavior scores.

Alzheimer's disease (AD) is a progressive neurodegenerative disease that often leads to dementia, characterized by loss of cognitive and memory function. Healthy patients are known as cognitively normal (CN), and clinicians have classified Alzheimer's disease into three distinct stages: the preclinical stage, an intermediate phase known as mild cognitive impairment (MCI), and, in the later stages, dementia of the Alzheimer's type (DAT) [26]. Preclinical AD affects the brain years before any diagnosis is made, and so there is a need to study brain changes in the early stages to aid detection and treatment methods. Unfortunately, there are often confounders or mixed pathologies, e.g., depression or Parkinson's disease [196], that complicate the diagnosis process. This places high importance on adaptable biomarkers that can differentiate overlapping pathologies such as fMRI, positron emission tomography (PET), and others. The first and most viable diagnostic marker, meaning the cheapest and easiest to implement, is one that does not require any imaging, such as a written cognitive exam that tests for things such as memory, language, and general cognition. The primary cognitive measure, also called a behavior score, analyzed in this chapter is the Montreal Cognitive Assessment (MoCA), which is designed for screening MCI subjects, and is superior to other exams such as the Mini-Mental State Examination (MMSE) [4]. The MoCA exam includes tasks in several areas, such as short-term memory recall, delayed recall, visual-spatial skills, executive function, sustained attention, language, and time/place orientation. MoCA can discriminate between MCI and CN subjects with an area under the curve (AUC) of 0.86, indicating the practicality of this metric for early diagnosis [4]. Additionally, in this chapter, we also compute average memory and language metrics for the prediction task. Incorporating these average subcategories with the MoCA metric provides finer, detailed information on the subject's cognitive performance.

7.1.2 Related Works

The intersection between behavior score prediction and imaging modalities is not entirely unprecedented. For example, the brain-behavior relationship has been studied in PET imaging using standardized uptake value ratio (SUVR) features [197]. More closely to this chapter, the association with MoCA and depression subjects has been studied in an rs-fMRI context [198] by using the functional connectivity predictive modeling (CPM) method [199]. In terms of behavior score prediction and AD subjects, Ref. [200] predicts the ADAS-Cog [201] metric using CPM on subjects spanning the AD spectrum. Further, Ref. [176, Chapter 2] does something similar with functional connectivity data using the CPM method in a cohort of purely MCI subjects. The Pearson correlation coefficient (R) for MoCA prediction is 0.07 in Ref. [176, Chapter 2] and 0.15 in Ref. [198], indicating a fairly weak correlation for MoCA prediction in rs-fMRI. It is worth noting that this is not a fair comparison, as these works use different rs-fMRI datasets and methodologies. However, these works consider only functional connectivity data, which collapses the time dimension and examines only brain region interactions. To the best of the authors' knowledge, there are no works that tackle behavior score prediction on the BOLD timeseries data itself, which may provide a richer context for finding more optimal predictions. Furthermore, these works do not investigate cognitive subcategories such as memory and language in this prediction problem, only broad metrics such as MoCA.

7.1.3 Contributions & Motivation

In recent brain stimulation research, Ref. [202] used high-definition transcranial direct current stimulation (HD-tDCS) to target the left ventrolateral prefrontal cortex exclusively in individuals with mild cognitive impairment (MCI). This intervention was administered in a double-blind protocol alongside either mnemonic strategy training or an autobiographical recall control condition over five consecutive daily sessions. The findings demonstrate pronounced neurophysiological effects during associative memory encoding in the experimental group relative to controls. These results underscore the importance of delineating specific brain regions implicated in cognitive impairment, thereby informing the potential of HD-tDCS applications in AD that extend beyond single-region stimulation paradigms.

In this chapter, we systematically contrast the rs-fMRI modality for predicting behavior scores in subjects spanning the AD spectrum using MoCA and subcategory metrics. This is done by exploring both functional connectivity and multivariate timeseries methods for this task, developing a data-driven deep learning method based on state space models that outperforms the other approaches, and drawing biological insights related to the brain-behavior relationship.

Category	Total Subjects	Scans (RS/FN/OL)	Age	Education	Race (W/B/A)	Sex (M/F)	MoCA
CN	202	202/276/64	71.6 ± 6.1	16.9 ± 2.1	168/31/3	131/71	27.1 ± 2.1
aMCI	53	53/74/16	73.3 ± 6.8	16.1 ± 2.2	44/8/1	16/37	23.4 ± 3.5
DAT	26	26/38/6	71.9 ± 6.2	15.9 ± 2.6	25/0/1	12/14	16.6 ± 5.0

Table 7.1: Distribution of subjects in the MADRC dataset, stratified by disease category.

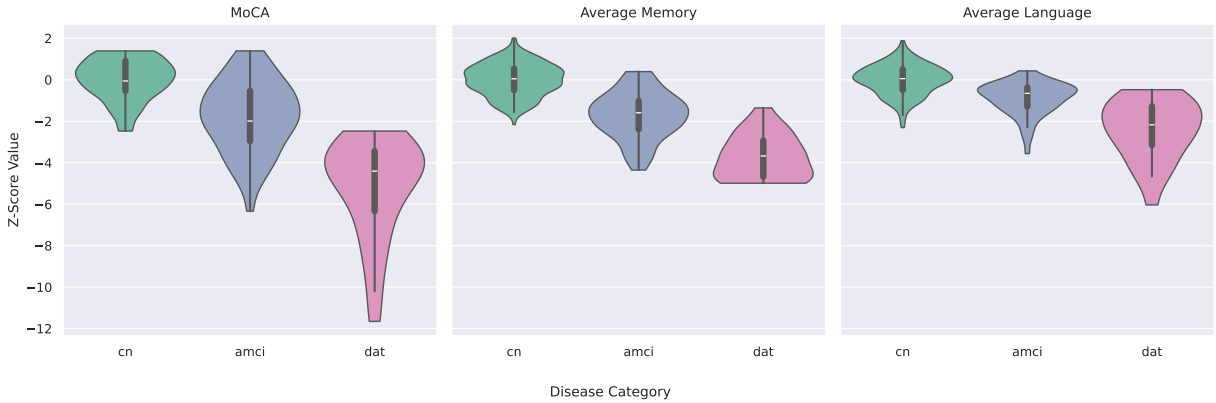


Figure 7.2: Violin plots illustrating the distribution of behavioral scores in z-score space across different disease categories.

7.2 Data Acquisition & Processing

7.2.1 Dataset

The Michigan Alzheimer's Disease Research Center (MADRC), in collaboration with the University of Michigan, has collected functional and structural 3T MRI single-session scans, along with behavioral score metrics, for use in this project as part of the National Institute on Aging (NIA) grant P30 AG072931. All participants provided written informed consent, and study activities were approved by the respective parties. Resting-state fMRI scans were completed with eyes open with a fixation cross. The 3T rs-fMRI data is acquired on a 32-channel Nova medical coil with a multi-band (MB) echo-planar imaging (MB-EPI) pulse sequence with MB factor of 6 leading to a 30ms echo time (TE), 0.8s repetition time (TR), a flip angle of $52°$, a 2.4mm spatial resolution, 60 total slices, and 570 total time samples.

The MADRC subject population consists of 281 subjects across all disease category labels. Generally, a subject can be cognitively normal (CN), have mild cognitive impairment (MCI), or dementia of the Alzheimer's type (DAT). MCI subjects that have memory-related impairment have the amnestic MCI (aMCI) label, while subjects that exhibit cognitive impairment (without memory-related impairment) have the non-amnestic MCI (naMCI) label. In the medical literature, this impairment grouping is common since aMCI subjects are more likely to develop into DAT subjects in the future. All MADRC MCI subjects used in this study are amnestic. For additional

details about the subject population, see Table 7.1 for demographic characteristics. Table acronyms are listed as follows: CN = cognitively normal, aMCI = amnestic mild cognitive impairment, DAT = Dementia of the Alzheimer's Type, W = White/Caucasian, B = Black/African American, A = Asian, M = Male, and F = Female. The behavior score distribution among subjects is illustrated in Fig. 7.2 by disease category and behavior score subcategories.

7.2.2 Behavior Score Metrics

As stated earlier, the MoCA score is the main cognitive metric used in this work. Additionally, memory and language submetrics are computed to provide a deeper understanding of cognitive impairment. In greater detail, a composite memory metric was calculated, averaging the z-scores of the UDS Benson complex figure recall [203], CRAFT story delayed recall [204], and CRAFT delayed verbal [204] behavioral tasks. A composite language metric was calculated, averaging the z-scores of total numbers of named animals [203], total number of named vegetables [203], and the Multilingual Naming Test (MINT) exam [55]. All metrics are z-score normalized using healthy population data.

7.2.3 Preprocessing

First, the raw data is processed by converting it to the standard BIDS format [184] for use in downstream software packages. Secondly, the application fmriprep [185] handles most standard tasks such as N4 bias field correction, skull stripping, normalization to the MNI152 linear space provided by the TemplateFlow archive [186], head-motion estimation, susceptibility distortion correction using fieldmaps, and z-score normalization of each voxel's timeseries data.

Secondly, the CONN toolbox [205] then applies a 6mm FWHM Gaussian smoothing kernel, detrending (quadratic), despiking (loess regression smoothing, piecewise cubic interpolation), nuisance regression (motion parameters by first derivative and quadratic modeling, cerebral spinal fluid and white components identified by PCA in masks), outlier scan removal, and bandpass filtering between 0.008-0.09Hz.

Lastly, the BOLD data is parcellated into 264 regions of interest (ROIs) using the Power atlas [206], with 8 additional ROIs we manually included ourselves to augment the atlas, since some areas of the amygdala and hippocampus may not be sufficiently covered, giving a total of 272 ROIs used in this project. The MNI coordinates and labels for these additional ROIs are included in Table 7.2. Please see github.com/javiersc1/NeuroMamba for additional information regarding our specific preprocessing pipeline.

ROI	MNI (x, y, z)	Label
265	(-24, -4, -20)	Left Amygdala
266	(24, -2, -20)	Right Amygdala
267	(28, -12, -20)	Right Hippocampus
268	(30, -24, -12)	Right Hippocampus
269	(30, -39, -3)	Right Hippocampus
270	(-29, -12, -22)	Left Hippocampus
271	(-30, -24, -12)	Left Hippocampus
272	(-29, -38, -4)	Left Hippocampus

Table 7.2: MNI coordinates and anatomical labels for additional regions of interest (ROIs) incorporated alongside the 264 ROIs defined in the Power atlas.

7.3 Formulation & Related Methods

7.3.1 Problem Formulation

For each subject i, a 3D volume is acquired over time to get BOLD data $V^i \in \mathbb{R}^{T \times S_X \times S_Y \times S_Z}$ where T represents time and (S_X, S_Y, S_Z) are spatial dimensions. After the preprocessing outlined in Sec. 7.2, the timeseries data becomes $X^i \in \mathbb{R}^{T \times B}$ where $B = 272$ refers to the brain region ROIs and T represents time samples. This timeseries data X^i will be the main focus of this work.

The goal then is to find some function $g(\cdot)$ that maps the given input, either X^i or some derivative of it, to the output $s^i \in \mathbb{R}^3$ where s^i represents the MoCA, memory, and language metrics for the ith subject. All subjects' scores are collected into a single matrix $S = [\, s^1, \ldots, s^N \,]$ where N denotes the number of samples. In this section, different methods are introduced that extract features from X^i to be used in a ridge regression model setting as discussed in Sec. 7.3.2.

7.3.2 Kernel Ridge Regression (KRR)

In standard ridge regression [207], the objective is to minimize the loss function with a regularization term to prevent overfitting by solving

$$\min_{W,B} \underbrace{\frac{1}{2}\|S - FW - B\|_{\mathrm{F}}^2}_{\text{data fidelity}} + \lambda \underbrace{\|W\|_{\mathrm{F}}^2}_{\text{prevent large values}} \tag{7.1}$$

where $\|\cdot\|_F$ is the Frobenius norm, W are the regression weights, B is the bias term, and $\lambda > 0$ is the regularization parameter. The input variable F consists of some features of the timeseries data $X = [\, X^1, \ldots, X^N \,]$. In (7.1), $g(F) = FW + B$ is the linear model assumption for F. To extend this approach, kernel ridge regression (KRR) [208] replaces the inner product of feature vectors with a kernel function $k(\cdot, \cdot)$, implicitly mapping inputs into a higher-dimensional space defined

by the kernel. The kernel matrix $K \in \mathbb{R}^{N \times N}$ is computed such that $K_{i,j} = k(F_{i,:}, F_{j,:})$. The KRR model optimizes the following objective

Kernel Ridge Regression (KRR)

$$\min_{W,B} \frac{1}{2}\|S - KW - B\|_{\mathrm{F}}^2 + \lambda \|KW\|_{\mathrm{F}}^2. \tag{7.2}$$

In our experiments, we use the radial basis function (RBF) kernel, also called the Gaussian kernel, defined as

$$k(F_{i,:}, F_{j,:}) = \exp(-\gamma \|F_{i,:} - F_{j,:}\|_2^2) \tag{7.3}$$

where $\gamma > 0$ is the kernel hyperparameter that controls the width or spread of the kernel. A larger γ means the kernel declines rapidly with distance, focusing more on close neighbors, while a smaller γ makes the kernel smoother and more global. The RBF kernel is the most widely used kernel function in machine learning because of its versatility and strong performance on a wide variety of tasks [209]. By using it, the model can capture nonlinear relationships between input features F and score metrics S, thereby improving predictive accuracy. From (7.2), it should be clear that different weights and bias terms are computed for each behavior score metric given the same input feature matrix F.

7.3.3 Connectivity-Based Methods

7.3.3.1 Functional Connectivity Matrix (FCM)

One can generate functional connectivity data by forming a Pearson product-moment correlation matrix from X^i. Let $C^i \in \mathbb{R}^{B \times B}$ represent the functional connectivity matrix. Mathematically, this is computed by

Functional Connectivity Matrix (FCM)

$$\forall j, k \in \mathcal{B}, \quad C^i_{j,k} = \frac{\mathrm{cov}(X^i_{:,j}, X^i_{:,k})}{\sigma(X^i_{:,j})\, \sigma(X^i_{:,k})} \tag{7.4}$$

where $\mathcal{B}$ is the set of brain regions, $\mathrm{cov}(\cdot)$ is covariance, and $\sigma(\cdot)$ is standard deviation. These connectivity values, C^i, are used as features in a kernel ridge regression setup by using (7.2).

7.3.3.2 Connectome Predictive Modeling (CPM)

Connectome predictive modeling (CPM) [199] is an approach to relate functional network strength to specific metrics. CPM has been applied in various studies of connectivity, such as predicting attention performance [195], individual identification [210], and fluid intelligence [211]. The method works by correlating connectivity data and behavioral measures. The resulting correlations are thresholded at a given level to determine significance. Next, the connectivity values at significant edges are summed separately for positive and negative correlations. This generates a "brain score" that is used in a linear regression setup.

Let $\tilde{C}^i = \mathcal{A}(C^i)$ represent the vectorized functional connectivity for the ith subject by applying (7.4). Let $\mathcal{C}_j$ represent a vector that consists of the jth edge values across the population. Let S_k represent a vector of the kth behavioral measure across the population. Then, the CPM method works by first generating a mask vector $M^k \in \mathbb{R}^{\tilde{B}}$ that selects the most significantly correlated edges, and summing the edge values for each subject. Mathematically, this can be expressed as

Connectome Predictive Modeling (CPM)

$$\begin{aligned} M_j^k &= \mathcal{T}^{+/-}(\text{corr}(\mathcal{C}_j, S_k)),\ \forall j \in \{1, \ldots, \tilde{B}\} \\ f_k^i &= \langle \tilde{C}^i \odot M^k, \mathbb{1}_{\tilde{B}} \rangle \end{aligned} \tag{7.5}$$

where $\text{corr}(\cdot, \cdot)$ is the correlation function, $\mathcal{T}^{+/-}(\cdot)$ is the operator that constructs a binary mask that comes from the most significantly correlated positive or negative edges, $\odot$ is element-wise multiplication, $\mathbb{1}_{\tilde{B}}$ is the ones vector of size $\tilde{B}$, and f_k^i is the brain score feature extracted for the ith subject associated with the kth behavior score. The matrix $F = [\, f^1, \ldots, f^N \,]$ is the input feature matrix used in ridge regression by solving (7.2).

7.3.4 Model-Based Timeseries Methods

7.3.4.1 Individual Independent Component Analysis (I-ICA)

Independent component analysis (ICA) [212] has become a fundamental tool in fMRI research for decomposing BOLD timeseries data into spatially independent components representing distinct functional networks [213]. ICA enables the identification of both task-based and resting-state networks without prior knowledge of temporal profiles, making it particularly valuable for exploratory analyses. To give a few examples, some common applications are the isolation of resting-state networks such as the default mode network [213], characterization of brain connectivity changes in neurological and psychiatric conditions [214], and the separation of physiological noise or motion

artifacts from true neural signals [215]. In our experiments, we use ICA to extract features from the multivariate timeseries data X^i.

ICA, similar to principal component analysis, is a method that finds a low-dimensional feature space consisting of independent sources/components and mixing coefficients that maximizes the mutual information between the feature space and the ambient space. The model assumes that $X = ME$ where M are the mixing coefficients and E are the independent sources/components. The goal is to find the inverse mapping $W = M^{-1}$ so that, given the original data X, one can recover the components by unmixing, i.e., $E = WX$. Mathematically, given timeseries data $\tilde{X} \in \mathbb{R}^{T \times B}$ that has zero mean and unit covariance, ICA is associated with the following objective

Independent Component Analysis (ICA)

$$\min_{W} \underbrace{\frac{1}{2}\|a(W\tilde{X})\|_F^2}_{\text{sparsity regularizer}} \text{ s.t. } \underbrace{WW' = I}_{\text{orthonormal sources}} \tag{7.6}$$

where W contains the inverse mixing coefficients, $W\tilde{X}$ are the components, a is a nonlinear convex function such as $\log(\cosh(\cdot))$, and $WW' = I$ is the independent constraint. This is done independently for each subject to extract components, and we denote this version as individual ICA (I-ICA). In this work, the components are features used in conjunction with KRR as the regression model by solving (7.2) on the input $\hat{W}\tilde{X}$ where $\hat{W}$ is the solution to solving (7.6) and $\tilde{X}$ is whitened version of X.

7.3.4.2 Group Independent Component Analysis (G-ICA)

In classical ICA for fMRI, labeled here as I-ICA, each subject's components are extracted independently, which can result in inconsistencies across subjects. The classical ICA method does not generalize to draw inferences about groups of subjects since different subjects will have different time courses, so it is not immediately clear how to extend the method for group data [216]. Thus, modifications are required to make the method more meaningful in a group context; many approaches have been developed for this in the fMRI context [216]. The most widely used group ICA method is a temporal concatenation approach known as GIFT/MELODIC [217], where a matrix $G \in \mathbb{R}^{TN \times B}$ is formed from temporally stacked data across all subjects. This ensures common sources are found across the population. Let $\mathcal{T}_i$ represent the set of all time samples associated with subject i and W_G is the result of solving (7.6) on $\tilde{G}$, a whitened version of G. Each subject contains a submatrix $W_{G_i} = W_G[\mathcal{T}_i, :]$ in W_G that contains the unmixing coefficients. Similarly as before, the components $\hat{W}_{G_i}\tilde{X}^i$ are extracted from W_G for all subjects and used in KRR as the regression model by solving (7.2).

7.3.4.3 Amplitude of Low Frequency Fluctuations (ALFF)

The amplitude of low-frequency fluctuations (ALFF) is a metric derived from BOLD timeseries data that extracts Fourier coefficients. ALFF quantifies the intensity of activity by measuring the power spectral density of low-frequency fluctuations for each brain region timeseries signal. Higher ALFF values indicate greater regional neural activity, providing insights into baseline brain function and potential alterations associated with neuropsychiatric conditions. To give a few examples, this method was used to predict mini mental state exam (MMSE) scores for subjects in the AD spectrum ($R = 0.21$) [218], predict frequency of migraines in individuals ($R = 0.35$) [219], and identify brain regions relevant in schizophrenia subjects [220]. Mathematically, for each subject i, this method computes

Amplitude of Low Frequency Fluctuations (ALFF)

$$a^i = \left[\mathcal{M}(|\mathcal{Z}(\mathcal{F}(X^i_{b,:}))|^{1/2}) \text{ for } b \in \{1, \ldots, B\}\right] \tag{7.7}$$

where $\mathcal{M}(\cdot)$ computes the mean, $\mathcal{Z}(\cdot)$ extracts the Fourier coefficients associated with frequencies 0.008-0.9Hz, $\mathcal{F}(\cdot)$ is the Fourier transform, and $|\cdot|$ computes the magnitude of the complex coefficients. The ALFF of all subjects is computed to form $A = [a_1, \ldots, a_N]^T \in \mathbb{R}^{N \times B}$. This ALFF-feature matrix, computed using (7.7), is used as the regression model in KRR by solving (7.2).

7.3.4.4 Fractional ALFF (fALFF)

Fractional ALFF (fALFF) [221] is a variant of ALFF that takes the sum of the low-frequency fluctuations in ALFF and divides by the total sum of all frequency magnitudes, including low and high frequencies. The traditional ALFF measure can be more sensitive to physiological noise, so this variant is known to improve the sensitivity and specificity in detecting brain activity [221]. Mathematically, for each ith subject, this variant computes

Fractional ALFF (fALFF)

$$\tilde{a}^i = \left[\frac{\mathcal{S}(|\mathcal{Z}_1(\mathcal{F}(X^i_{b,:}))|^{1/2})}{\mathcal{S}(|\mathcal{Z}_2(\mathcal{F}(X^i_{b,:}))|^{1/2})} \text{ for } b \in \{1, \ldots, B\}\right] \tag{7.8}$$

where $\mathcal{S}(\cdot)$ computes the sum, $\mathcal{Z}_1(\cdot)$ extracts the Fourier coefficients associated with frequencies 0.008-0.9Hz, and $\mathcal{Z}_2(\cdot)$ extracts the Fourier coefficients associated with frequencies 0.0-0.25Hz.

The feature matrix is computed to form $F = [\, \tilde{a}_1, \ldots, \tilde{a}_N \,] \in \mathbb{R}^{N \times B}$. The matrix F is the input to the ridge regression model by solving (7.2).

7.3.5 Data-Driven Timeseries Methods

7.3.5.1 Temporal Convolutional Network (TCN)

Temporal Convolutional Networks (TCNs) [222] are deep models designed specifically for processing sequential data. They are based on convolutional neural networks (CNNs), which are well-known for their efficiency in processing visual data such as images. In TCNs, the convolutional layers are applied along the time dimension of the sequential data. This model is adapted for regression by changing the head to instead perform global average pooling along the time dimension and adding a linear layer to predict behavior scores from the number of channels or variates in this instance.

7.3.5.2 Long Short-Term Memory (LSTM)

Long short-term memory networks (LSTMs) [223] are a specialized type of a recurrent neural network (RNN) designed to address the vanishing and exploding gradient problems commonly encountered in standard RNNs. By incorporating gated memory cells, LSTMs can effectively capture and utilize long-range dependencies in sequential data, making them highly successful across a variety of tasks. This is due to their use of input, output, and forget gates, which allow the model to selectively retain relevant information over extended sequences. Since their introduction, LSTMs have served as a foundational building block for many advances in sequence modeling and have inspired a range of related architectures. In our experiments, a bidirectional variant [224] of the LSTM model is used and denoted as "BiLSTM". This architecture is adapted for regression, similar to the TCN model.

7.3.5.3 Patch Time Series Transformer (PatchTST)

The Patch Time Series Transformer (PatchTST) [225] adapts the Transformer architecture [226] for time series tasks by introducing a patch-based input representation. Unlike traditional approaches that operate on individual time steps, PatchTST partitions the input sequence into overlapping patches, enabling the model to capture both local and global patterns efficiently. This design leverages self-attention mechanisms to model complex temporal dependencies across multiple variables. PatchTST has achieved excellent performance on multivariate timeseries forecasting tasks [225], demonstrating superior performance compared to classical Transformer models. This architecture is adapted for regression, similar to the TCN model.

7.4 Proposed Method

7.4.1 Deep State Space Models (SSMs)

State space models (SSMs) model physical systems using state variables that track how inputs change system behavior over time and can be written in matrix form for linear, time-invariant (LTI) systems [227]. Given an input signal $x(t) \in \mathbb{R}$, the SSM will generate an output sequence $y(t) \in \mathbb{R}$ using the state $h(t) \in \mathbb{R}^L$ by equations

$$\begin{aligned} h'(t) &= Ah(t) + Bx(t) \\ y(t) &= Ch(t) + Dx(t) \end{aligned} \tag{7.9}$$

where A is the state matrix, B is the input matrix, C is the output matrix, D is the feed-through matrix, and L is the state size. In well-known physical problems, the matrices $\{A, B, C, D\}$ are determined by differential equations or by physics-based modeling of the system. In contrast, for complex systems, the dynamics are either unknown or difficult to model, so the $\{A, B, C, D\}$ matrices are learned in a data-driven way in the deep SSM context. The first popular work that studies this deep SSM idea and addresses some important challenges, such as matrix initialization and fast parallelized computation, is the structured SSM (S4) model [228]. Since acquired data is discrete with some time step Δ, the recurrence equation in (7.9) is rewritten using zero-order hold discretization [229] by

S6 Block

$$\begin{aligned} &\bar{A} = \exp(\Delta A), \quad \bar{B} = (\Delta A)^{-1}(\exp(\Delta A) - I) \cdot \Delta B, \\ &y_k = Ch_t, \quad h_t = \bar{A}h_{t-1} + \bar{B}x_t. \end{aligned} \tag{7.10}$$

Note that D is dropped since it can easily be implemented as a skip connection in a neural network context. Note that both (7.9) and (7.10) are for a single sequence, and so to process multiple variates, multiple independent SSM instances are used in practice.

However, since the SSM modeling in (7.9) and (7.10) are time-invariant, this poses challenges for certain tasks such as noun selection in language problems. Thus, the S6 layer, named after S4 combined with selective scanning, employs a selective mechanism in which $\{A, B, C\}$ are time-varying, making the SSM content-aware [229, 230]. However, since the system is no longer LTI, the update rules for $\{A, B, C\}$ can no longer be easily parallelized in FFT multiplication form. Instead, the recurrence equation is used, but it is still efficiently parallelized using parallel scan

methods, also known as parallel prefix sum algorithms [231]. Additionally, the S6 layer remains linear, so in practice the layer is wrapped around a "Mamba block" formalized by

Mamba Layer

$$H = \sigma(\text{Conv1D}(\text{Affine}(X))), \quad Z = \sigma(\text{Affine}(X)),$$
$$Y = \text{S6}(H), \quad \hat{Y} = \text{Affine}(Y \odot Z) \tag{7.11}$$

where X is the input, $\hat{Y}$ is the output, σ is SiLU activation, $\text{S6}(\cdot)$ is the operation in (7.10), and $\odot$ is element-wise multiplication with the gating mechanism Z. This time-varying S6 layer, wrapped around nonlinear components, enables a non-LTI SSM formulation that better captures the nuances of complex systems that do not satisfy linearity or time-invariance. However, this Mamba model was primarily designed for language tasks and therefore does not leverage domain-specific knowledge, such as multivariate timeseries data. In Sec. 7.4.2, modifications to the Mamba model are proposed, such as bidirectionality and differential design, to aid with modeling temporal dynamics in the timeseries data.

7.4.2 NeuroMamba

7.4.2.1 Bidirectionality

To enhance Mamba's ability to preserve historical information over a longer time range, we generalize by introducing two Mamba blocks: one to capture forward temporal dynamics and another to capture backward temporal dynamics. This addresses the limitation of SSMs, which forget past information at longer horizons [232], by focusing on both recent and past information. We note that this kind of idea is not novel, as similar ideas have already been proposed for long short-term memory networks (LSTMs) and others [224]. The Mamba model does not have this functionality because it was designed for language tasks where the goal is to auto-regressively predict the next token in a left-to-right fashion. However, in this fMRI application, one is more interested in learning from temporal dynamics to predict behavior scores, and because this is not a model for real-time use, bidirectionality enables a more predictive model in this offline context.

7.4.2.2 Differential design

Sequence models such as Transformers and RNNs tend to over-allocate attention to irrelevant context, especially for multi-needle retrieval language tasks where the goal is to answer questions embedded in a pile of documents. Recent works have focused on differential attention [233] to calculate the difference between two attention mechanisms, and this was found to cancel noisy

intermediate representations, similar to differential amplifiers in electrical engineering. This led to advantages in key areas such as information retrieval, hallucination mitigation, in-context learning, and the reduction of activation outliers [233]. This idea has been recently extended to RNNs and SSMs [234]. We incorporate this idea because of similarities to our problem, e.g., finding a few variates embedded within the entire list that are most impactful for prediction, similar to multi-needle retrieval. Because this differential design never performed worse than the baseline model [233], we integrate bidirectionality to instead compute the learnable scaled difference rather than simply adding the two Mamba blocks.

7.4.2.3 Small Batch Regularization (SBR)

Neural networks with strong performance are almost always over-parameterized, which is why they are called "deep". Despite explicit regularization to avoid overfitting, some studies have found that the implicit regularization of small-batch training tends to converge to minima with better generalization performance, regardless of whether explicit regularization is used [235, 236, 237]. Specifically, with limited training samples, the regularization effect remains strong [235], and training with multiple dataset passes, also called epochs, is theoretically optimal relative to single-pass training [237]. We incorporate small-batch regularization (SBR) to train our model and achieve better generalization performance while preventing overfitting on our limited-sample dataset.

7.4.2.4 Model Overview

Putting everything together, the bidirectionality and differential design ideas are integrated to form an updated "Mamba++" layer utilized in the NeuroMamba model. This layer is described by taking some input feature F from the last layer and performing the following

Mamba++ Layer

$$\begin{aligned} \tilde{F}_F &= \text{Mamba}(F), \quad \tilde{F}_B = \mathcal{T}(\text{Mamba}(\mathcal{T}(F))) \\ F_O &= \lambda_\theta^2 \odot \text{RMSNorm}(\lambda_\theta^1 \odot \tilde{F}_F - \tilde{F}_B) + F \end{aligned} \tag{7.12}$$

where $\lambda_\theta^1, \lambda_\theta^2 \in \mathbb{R}^B$ are learnable scaling parameters, RMSNorm$(\cdot)$ is root mean square normalization [238], and $\mathcal{T}(\cdot)$ is the linear operator for time flipping. Once the sequence has been processed by the Mamba++ layers, it is temporally averaged to summarize session statistics for every brain region. This forms a latent vector $h \in \mathbb{R}^B$ that is fed to an affine layer for score prediction. Let $f_\theta(\cdot)$ denote the NeuroMamba backbone that consists of a stack of Mamba++ layers combined

with the temporal average pooling linear operator $\mathcal{P}(\cdot)$, $g_\theta(\cdot)$ denote the readout function that is simply an affine layer for linear regression, and θ correspond to the set of learnable parameters. Then, the NeuroMamba model is summarized by

NeuroMamba

$$f_\theta(\cdot) = \mathcal{P}([\text{Mamba++}_\theta^{(l)}(\cdot) \text{ for } l = 1 \dots L]) \tag{7.13}$$

$$g_\theta(\cdot) = W_\theta f_\theta(\cdot) + b_\theta \tag{7.14}$$

$$\min_\theta \sum_{i=1}^{N} \frac{1}{2} \underbrace{\|s - g_\theta(f_\theta(X^i))\|_2^2}_{\text{data fidelity}} + \lambda \underbrace{\|f_\theta(X^i)\|_1}_{\text{sparse brain regions}} \tag{7.15}$$

where (7.15) is the cost function used to find optimal model parameters, s contains the behavior score metrics, and $\|\cdot\|_1$ is an L1 penalty term to promote a sparse number of brain regions that are relevant for prediction. See Fig. 7.1 for a pictorial representation of the NeuroMamba model. The code associated with all of these methods and experiments in this chapter is published at github.com/javiersc1/NeuroMamba. In Sec. 7.5, the experimental setup and results of these methods are discussed.

7.5 Results & Discussion

7.5.1 Setup

Due to the limited size of our dataset and the need to ensure generalization across the full distribution, we adopt a leave-one-out approach. Specifically, for each fold, the methods are trained on all data except one sample, which is held out for testing. This practice is commonly used in related fMRI prediction studies [176, 200, 198]. The {FCM, CPM, I-ICA, G-ICA, ALFF, fALFF} algorithms perform feature extraction by optimizing a convex cost function, and subsequently employ kernel ridge regression (KRR), which is also a convex optimization. Consequently, these methods are run until the optimization process converges. In contrast, the deep learning approaches {TCN, BiLSTM, PatchTST, NeuroMamba} involve non-convex optimization and are therefore trained for a fixed number of 50 epochs, as further epochs yield minimal improvements in the cost function at this stage for all methods. All relevant hyperparameters for the various methods were determined via cross-validation using a randomly selected training/validation split of $50\%/50\%$, ensuring that the chosen parameters are representative of the overall dataset. To promote fair assessment of model generalizability, these parameters remain fixed across all folds and are not re-optimized for each individual fold.

Method	MoCA	Memory	Language
FCM + KRR	0.07	0.08	0.10
CPM_{pos} + KRR	0.06	0.01	0.08
CPM_{neg} + KRR	0.05	0.03	0.11
I-ICA + KRR	0.14^{*}	0.09	0.10^{*}
G-ICA + KRR	0.18^{**}	0.12^{*}	0.16^{**}
ALFF + KRR	0.20^{***}	0.13^{*}	0.18^{**}
TCN	0.26^{***}	0.16^{**}	0.17^{**}
BiLSTM	0.19^{***}	0.19^{***}	0.18^{**}
PatchTST	0.28^{***}	0.17^{**}	0.19^{***}
NeuroMamba	$\mathbf{0.36}^{***}$	$\mathbf{0.24}^{***}$	$\mathbf{0.25}^{***}$

Table 7.3: Pearson correlation coefficients (R) and corresponding p-values (p) by score category for multiple methods applied to the MADRC rs-fMRI data. Asterisks denote statistical significance as follows: $* = p < 0.1$, $** = p < 0.01$, $*** = p < 0.001$.

7.5.2 Experiments

7.5.2.1 Predictive Accuracy

In this experiment, the accuracy of different approaches is compared by analyzing the Pearson correlation coefficient (R) and associated p-values between the predicted and true behavior scores for each cognitive subcategory. In Table 7.3, the R values are reported for MoCA, average memory, and average language for each method; for brevity, only the MoCA R values are discussed below. As demonstrated, FCM achieves $R = 0.07$, which closely aligns with published literature values of $R = 0.07$ [176] and $R = 0.15$ [198]. This is not a fair comparison given the different datasets and methods employed, yet our value appears in line with these works. Further, the CPM method achieves a similar range with the positive edge model achieving a value of $R = 0.06$, which is not too different from FCM.

Regardless, this is where the literature ends, with functional connectivity. Compared to the timeseries approaches that perform better, this indicates that important temporal dynamics useful for behavior score prediction are lost. Shifting to model-based timeseries methods, I-ICA improves over the FCM baseline by learning the distinct sources that compose the BOLD data, leading to a MoCA value of $R = 0.14$. However, since the sources are independent across subjects, this individual approach lacks group-level information that would enable shared sources among individuals. Thus, the G-ICA method with $R = 0.18$ provides an advantage in finding shared sources at the population level. Yet, the ALFF approach turns out to be the best *model-based* method, with $R = 0.20$.

However, these previous approaches use *model-based* feature extraction that may not be optimally relevant for behavior score prediction. By comparison, the deep learning methods performed

Method	MoCA	Memory	Language
Mamba	0.23^{***}	0.12^{*}	0.19^{***}
+bidirectionality	0.28^{***}	0.18^{**}	0.19^{***}
+differential	0.34^{***}	0.24^{***}	0.22^{***}
+SBR = NeuroMamba	$\mathbf{0.36}^{***}$	$\mathbf{0.24}^{***}$	$\mathbf{0.25}^{***}$

Table 7.4: Comparative ablation analysis illustrating the performance differences between NeuroMamba and the standard Mamba architecture.

relatively well with values within $R = 0.19 - 0.28$. Shifting to NeuroMamba, our *data-driven* multivariate timeseries method, it achieved the highest value $R = 0.36$, indicating superior capability in learning temporal dynamics.

7.5.2.2 Ablation Study

The ablation analysis, shown in Table 7.4, compares NeuroMamba relative to the standard Mamba architecture. These results reveal progressive improvements in predictive accuracy across cognitive categories. The bidirectional component with non-causal design leads to a 22% improvement for MoCA relative to Mamba by better capturing long-range dependencies and patterns in the complex data. Further, the differential component leads to a 11% improvement over the bidirectional Mamba variant by amplifying "attention" to related brain regions relevant to prediction. Finally, using small batch regularization (SBR) as a training technique leads to a 10% improvement over the differential variant, demonstrating its use to prevent overfitting by introducing noisy gradient updates that improve generalization.

7.5.2.3 Behavior Score Plots

The predicted behavior scores are computed for the entire dataset and plotted against the true behavior scores in Fig. 7.3 for the NeuroMamba model with reported (R, p)-values for each behavior score and disease categories. Across the behavior scores, the highest correlation values are observed in the amnestic MCI group, suggesting its potential as a marker for at-risk populations that may develop into late-stage dementia of the Alzheimer's type. For unknown reasons, the correlation values are not as strong for CN and DAT individuals as indicated in Fig. 7.3.

7.5.2.4 Impactful Brain Regions

The NeuroMamba backbone $\mathcal{P}(f_\theta(X^i)) = h^i \in \mathbb{R}^B$ extracts B elements, one element for each brain region in the list of Power atlas ROIs. This is fed to an affine layer $g_\theta(h^i) = W_\theta h^i + b_\theta$ to pick out some sparse combination of brain regions that are useful for prediction. In a standard linear

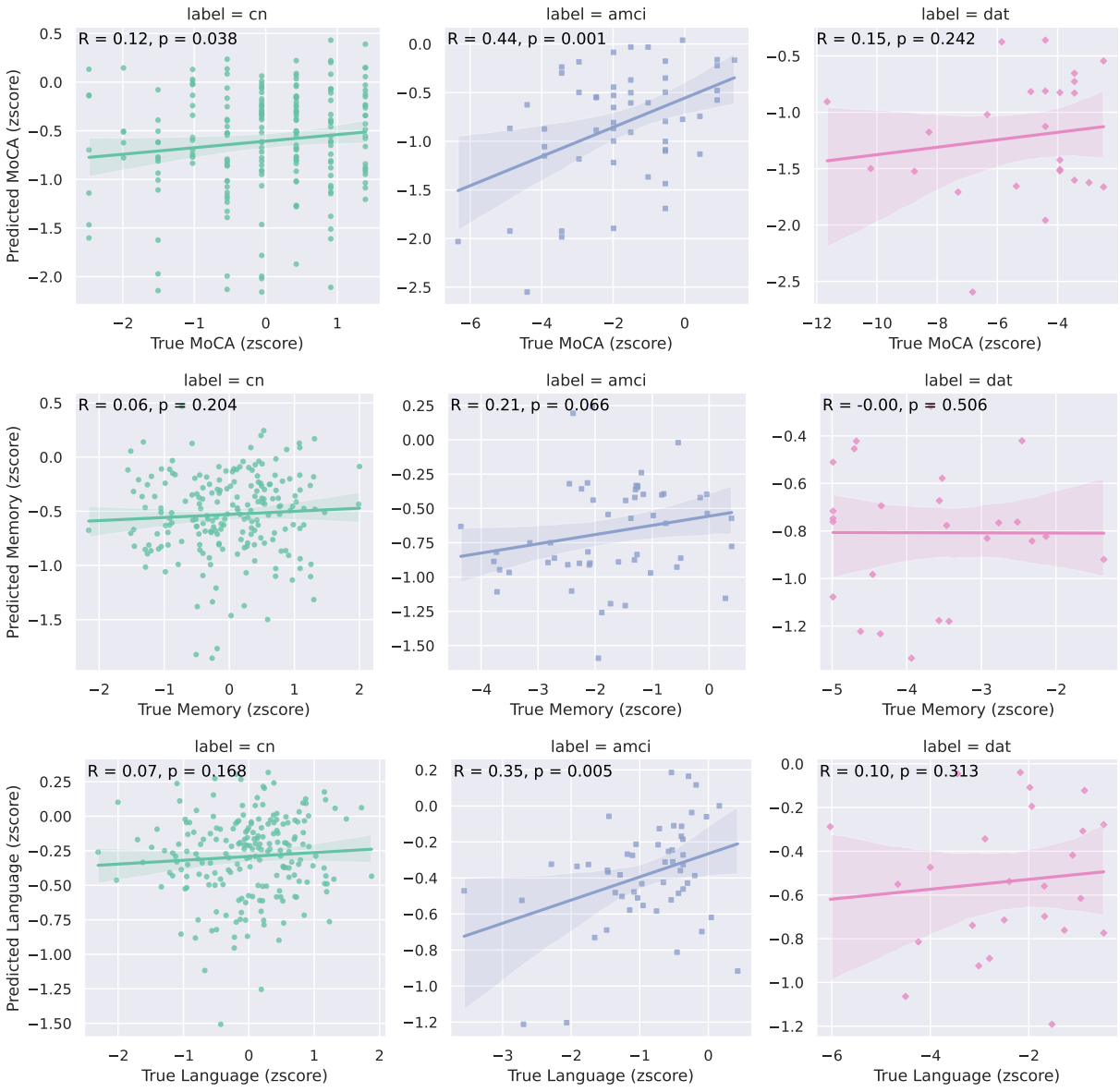


Figure 7.3: Correlation scatter plots displaying the relationship between predicted NeuroMamba scores and true behavioral metrics (rows), across the Alzheimer's disease spectrum (columns), presented in z-score normalized space.

PFI (RMSE)	Power ROI	MNI (x,y,z)	Nominal System	Lobe/Area	Talairach Daemon Label
			MoCA		
1.84 ± 0.05	77	(-13, -40, 1)	Default Mode	Limbic Cortex	Parahippocampal Gyrus
0.39 ± 0.02	170	(6, -81, 6)	Visual	Occipital Lobe	Cuneus
0.21 ± 0.02	177	(-53, -49, 43)	Fronto-parietal Task Control	Parietal Lobe	Inferior Parietal Lobule
0.07 ± 0.01	212	(-11, 26, 25)	Salience	Limbic Cortex	Anterior Cingulate
0.05 ± 0.01	93	(15, -63, 26)	Default Mode	Occipital Lobe	Precuneus
			Memory		
1.62 ± 0.05	77	(-13, -40, 1)	Default Mode	Limbic Cortex	Parahippocampal Gyrus
0.27 ± 0.02	177	(-53, -49, 43)	Fronto-parietal Task Control	Parietal Lobe	Inferior Parietal Lobule
0.26 ± 0.01	111	(-11, 45, 8)	Default Mode	Limbic Cortex	Anterior Cingulate
0.22 ± 0.01	170	(6, -81, 6)	Visual	Occipital Lobe	Cuneus
0.15 ± 0.01	93	(15, -63, 26)	Default Mode	Occipital Lobe	Precuneus
			Language		
0.79 ± 0.02	77	(-13, -40, 1)	Default Mode	Limbic Cortex	Parahippocampal Gyrus
0.60 ± 0.01	221	(2, -24, 30)	Emotion/Behavior	Limbic Cortex	Cingulate Gyrus
0.30 ± 0.01	170	(6, -81, 6)	Visual	Occipital Lobe	Cuneus
0.26 ± 0.01	177	(-53, -49, 43)	Fronto-parietal Task Control	Parietal Lobe	Inferior Parietal Lobule
0.26 ± 0.01	19	(13, -33, 75)	Sensory/Motor	Frontal Lobe	Precentral Gyrus

Table 7.5: Top five brain regions implicated in behavior score prediction, ranked by importance using permutation feature importance (PFI) for each score category. Additional columns provide MNI coordinates, nominal system category, lobe classification, and Talairach Daemon (TD) labels.

regression problem where all features are z-score normalized, one can use the learned magnitude weights in W to directly rank the most impactful features. However, in the deep learning context, this condition is not necessarily met, necessitating more advanced approaches to identify impactful features.

In this work, permutation feature importance (PFI) [239] is used to aid in this problem. PFI is a method for assessing the significance of features in a black-box model by measuring how much the prediction error increases when the values of a single feature are randomly shuffled across samples. If shuffling a feature significantly reduces the model's accuracy, that feature is considered important for the prediction task. This is done for each element $h \in \mathbb{R}^B$ independently across 100 shuffling trials, while holding the other features fixed. The PFI method is applied for each behavior score category to identify relevant brain regions. The root mean square error (RMSE) metric is used to measure model performance degradation, and the top 5 brain regions for each behavior score are shown in Table 7.5. Additionally, the Power atlas ROI index, MNI-space coordinates, nominal system, lobe/area, and Talairach Daemon labels [240] are included for each brain region to enhance discussion.

For brevity, only MoCA-related brain regions are discussed here. The top 5 brain regions selected are the parahippocampal gyrus, cuneus, inferior parietal lobule, anterior cingulate, and precuneus. The parahippocampal gyrus is important for memory encoding, retrieval, spatial navigation, and contextual processing [241]. For AD subjects, the parahippocampal gyrus shows atrophy with decline in episodic memory [242], reduced activation during memory tasks in fMRI [243], early deposition of amyloid-beta plaques and tau proteins [244], and is considered a key node in the DMN with disrupted connectivity [30]. Because of this, it is not surprising that this region had a significant impact. The cuneus, a region of the occipital lobe responsible for higher-order visual processing, has been implicated in cognitive impairment through atrophy and hypometabolism associated with deficits in visual processing, spatial navigation, and attentional functions in AD subjects [46, 245, 246]. The precuneus is a region with many daydreaming-like functions, such as mental imagery, episodic memory, and self-reflection, and it is an active part of the DMN in rs-fMRI [247]. Similarly, the precuneus shows reduced metabolism and atrophy in AD subjects [248, 249]. The inferior parietal lobule (IPL) is involved in various cognitive processes, including speech, language, spatial reasoning, working memory, and number processing [250]. The IPL undergoes changes in thickness of its banks during the transition from healthy to mild impairment [251], and abnormal functional connectivity changes occur relative to other networks, such as salience, sensorimotor, and executive networks [252]. Lastly, the anterior cingulate (ACC) plays a major role in emotion regulation and processing, like impulse control, motivation, goal-directed behavior, and emotional pain perception [253]. For AD subjects, the ACC exhibits reduced thickness [254] and decreased functional connectivity [255], which has connections to high agitation, irritability, and anxiety in people with AD [256]. To summarize, our findings of relevant brain regions for behavior score prediction in rs-fMRI closely align with established medical research on impaired regions in AD, which may be useful for intervention strategies such as multi-region HD-tDCS applications.

7.5.2.5 Saliency Maps

In deep learning, a saliency map [257] is a visual tool that highlights which parts of the input signal are most important for a model's prediction. In this context, given a subject's timeseries data $X^i \in \mathbb{R}^{T \times B}$ and the predicted MoCA score, one can backpropagate the gradients of the model back to the input space to analyze which timeseries blocks in T were important. In Fig. 7.4, a mildly impaired subject's BOLD timeseries data is shown with the top 5 most impactful regions identified in Sec. 7.5.2.4. The NeuroMamba model predicted a MoCA score that closely aligned with the true score for this individual. Here, brighter colors such as red and orange indicate timeseries portions that were useful for score prediction. The model does not enforce a global constraint of time points to ignore; however, one can see similar time regions selected across brain regions. This

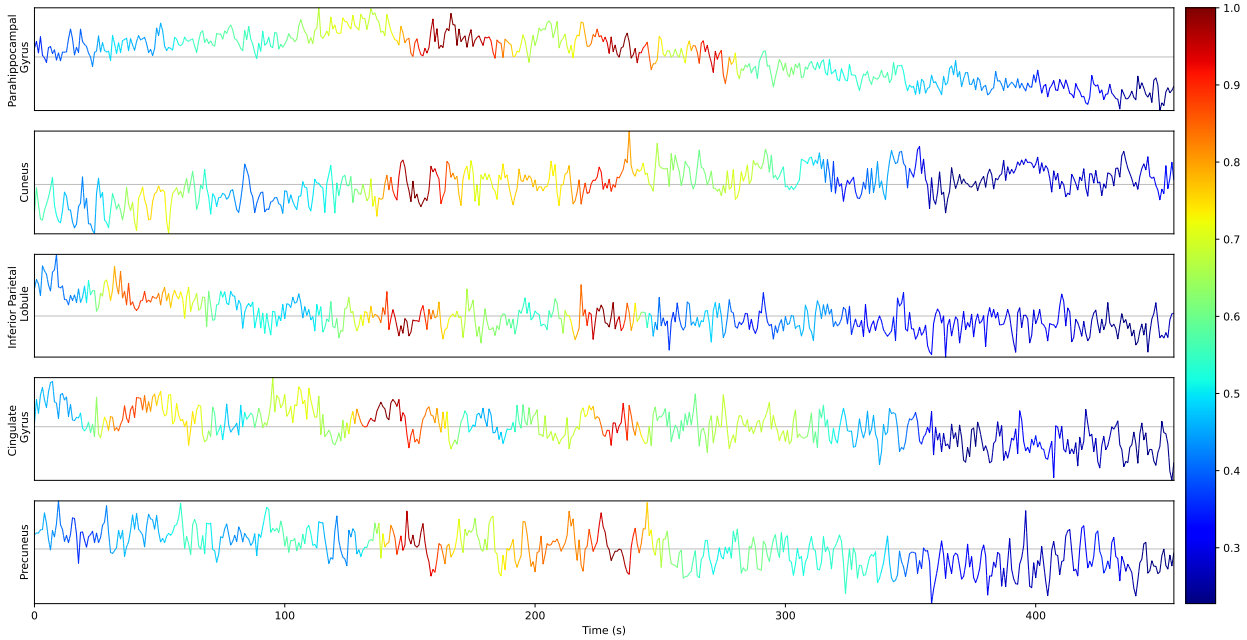


Figure 7.4: BOLD time series data from a single subject, highlighting a subset of regions identified by NeuroMamba. Specifically, from top to bottom, the parahippocampal gyrus, cuneus, inferior parietal lobule, cingulate gyrus, and precuneus. Color represents the strength of saliency and thus the importance of each time segment for MoCA score estimation.

suggests and supports the idea of spontaneous activity in rs-fMRI, especially in areas associated with the DMN. However, it is impossible to know, in general, whether a subject was actually daydreaming/self-referential processing in practice. It would be interesting to compare this result with task-based fMRI, which provides associated timing information, meaning one knows whether a subject is performing a task or at rest.

7.5.2.6 Early Biomarker Feasibility

There have been many works that explore classification/diagnosis of subjects in the AD spectrum using rs-fMRI data [16, 258, 259, 260, 261, 262]. They all follow the same general formula: 1) introduce the idea that functional changes occur in the brain earlier than structural ones meaning there is promise for early diagnosis, 2) develop some novelty in the methodology, e.g., going from a convolutional to a transformer-based network, that performs relatively better for diagnosis, and 3) reiterate that these results indicate the potential for rs-fMRI data to be used as an early biomarker for diagnosis. However, these works and many others miss the mark. The more interesting question is whether rs-fMRI data adds something useful to diagnosis that is not already provided by easily acquired metrics such as MoCA. In this experiment, we aim to analyze the classification performance of {only using MoCA, MoCA with FCM features, MoCA with G-ICA features, and MoCA

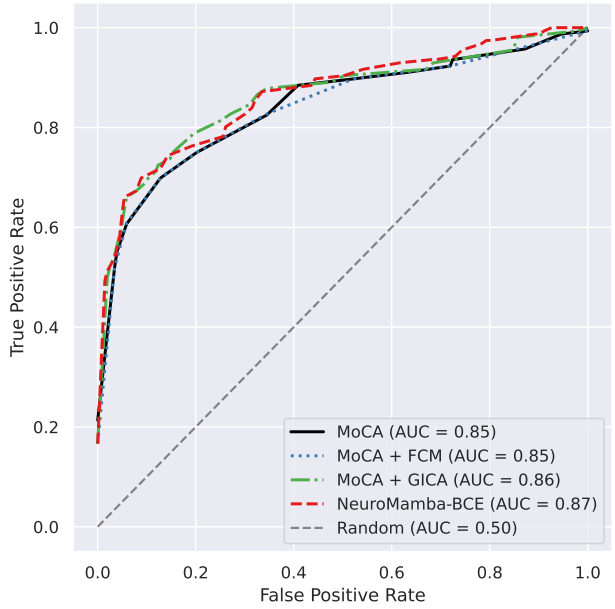


Figure 7.5: Receiver operating characteristic (ROC) curve with area under curve (AUC) values for diagnosis between cognitively normal and non-normal subjects using MoCA score in conjunction with rs-fMRI features.

with the proposed NeuroMamba model}. For the model-based approaches, logistic regression is used to predict the probability of belonging in the positive class (aMCI and DAT) from the negative class (CN). For NeuroMamba, a variant named NeuroMamba-BCE is created that combines the backbone model f_θ with MoCA values s by

NeuroMamba-BCE

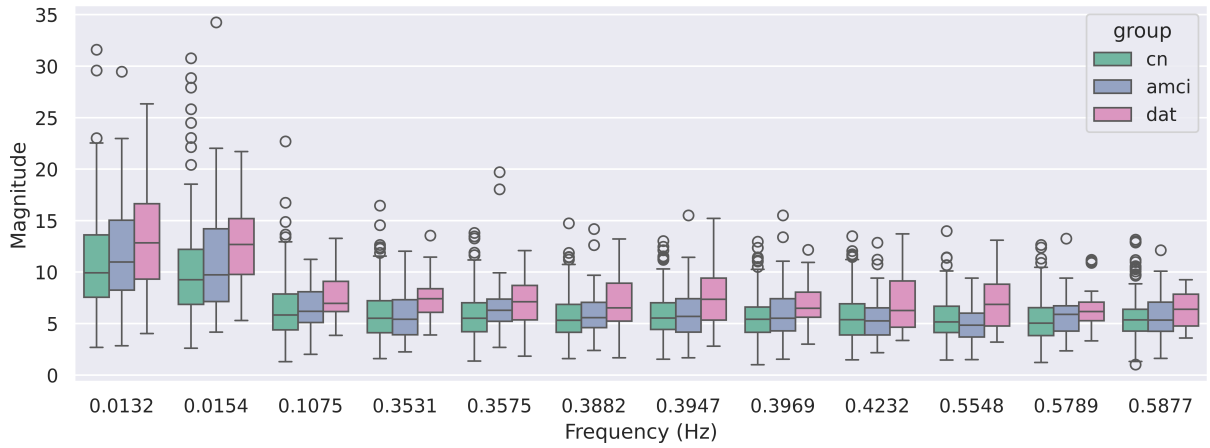
$$f_\theta(\cdot) = \mathcal{P}([\text{Mamba++}_\theta^{(l)}(\cdot) \text{ for } l = 1 \dots L]) \tag{7.16}$$

$$g_\theta(\cdot, s^i) = W_\theta \, \text{Concat}[\underbrace{f_\theta(\cdot)}_{\text{rs-fMRI features}}; \overbrace{s^i}^{\text{MoCA}}] + b_\theta \tag{7.17}$$

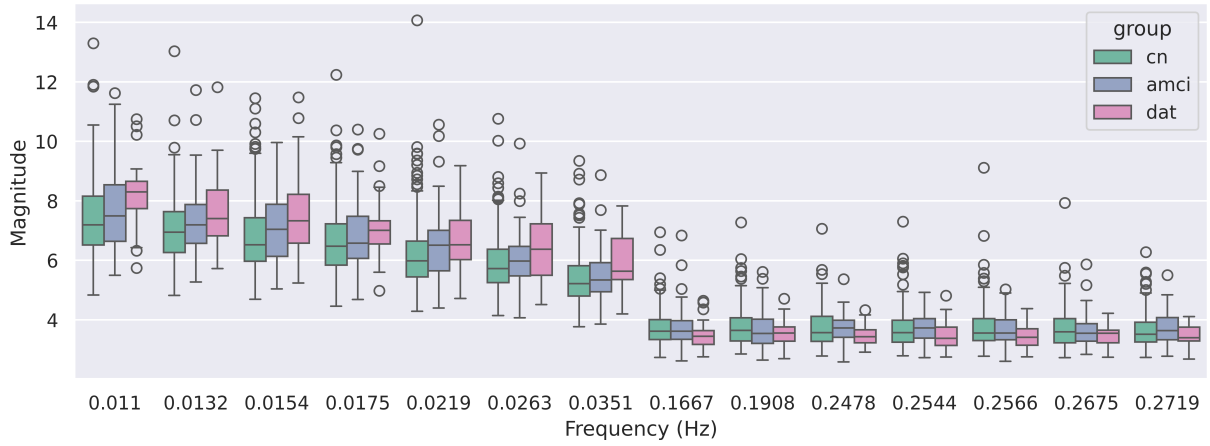
$$\min_\theta \sum_{i=1}^{N} \underbrace{\mathcal{L}_{\text{BCE}}(g_\theta(X^i,\ s^i);\ c^i)}_{\text{binary cross entropy}} + \lambda \underbrace{\|f_\theta(X^i)\|_1}_{\text{sparsity reg.}} \tag{7.18}$$

and train the model using the binary cross entropy loss function to learn the mapping between features $\{X^i, s^i\}$ and subject label c^i. Since it directly predicts probabilities, logistic regression is not needed here in this deep approach.

The receiver operating characteristic (ROC) curves for these approaches are plotted in Fig. 7.5, along with the corresponding area under the curve (AUC) values. From these results, there was



(a) Frequency spectrum of parahippocampal gyrus (node=247) at statistically significant frequencies.



(b) Frequency spectrum averaged over all nodes at statistically significant frequencies.

Figure 7.6: Frequency spectrums of resting-state MADRC fMRI data at frequency locations where sample means between CN and DAT classes are statistically different ($p \leq 0.05$).

no improvement in diagnostic ability across connectivity and time series-based methods, including our proposed model. This negative result indicates that *resting-state* fMRI is unlikely to provide additional diagnostic power beyond that already provided by MoCA and other readily available metrics. However, it is possible that *task-based* fMRI could provide more complementary information, given it is more likely to better "stress" the brain networks. Further research is necessary to conclusively determine whether fMRI as a whole is beneficial for early diagnosis.

7.5.2.7 Frequency Analysis

Since ALFF performs reasonably well to predict MoCA scores in Table 7.3, one can argue that there must be some frequencies that reasonably explain increased or decreased activation in certain brain regions that predict MoCA performance. Taking a closer look across the disease categories,

Fig. 7.6 shows the frequency spectrum of resting-state fMRI data at frequency locations where the sample means between the CN and DAT classes are statistically significant ($p \leq 0.05$). This is done for a specific brain region, the parahippocampal gyrus, and averaged over all brain regions to see if there is a general trend of frequency fluctuations between disease categories. As shown in Fig. 7.6, in both subplots, DAT subjects exhibit higher magnitudes overall across frequencies, indicating possible disrupted metabolic activity at these locations. With some ultrasound-focused brain stimulation techniques, it is possible to excite or inhibit metabolism at localized brain regions depending on the stimulation frequency [263]. Thus, a greater understanding of disrupted metabolism in AD helps plan brain stimulation design protocols.

7.5.2.8 Out of Distribution (OOD) Generalization

The Alzheimer's Disease Neuroimaging Initiative (ADNI) [2] began in 2004, with the main goals of providing data to researchers and improving doctors' diagnoses of subjects with AD. Part of this data collection includes the MoCA exam and 3T rs-fMRI EPI data (Flip Angle=90°, TE=30ms, TR=3.0s, 3.4mm spatial resolution, 192 total time samples), which allows us to explore how these methods generalize to a different dataset. The ADNI dataset is prepared using a methodology similar to that in Sec. 7.2.3, with a few key differences. Namely, fieldmap data is inconsistent and missing for some subjects, so fieldmap-less susceptibility distortion correction is done instead. Additionally, since the TR is quite high, slice timing correction is needed to ensure that all slices are treated as if they were acquired simultaneously. The MoCA scores are z-score normalized per the MADRC normal population. The ADNI data consists of 471 subjects (324 CN, 110 aMCI, 37 DAT) over 860 sessions (579 CN, 220 aMCI, 61 DAT). Care is taken to ensure no leakage during testing, i.e., a subject's scans are held out entirely rather than session only.

In this section, we test for two things: zero-shot transfer and all-shot training. Zero-shot means a method is trained on MADRC and tested on ADNI using the same hyperparameters from MADRC. All-shot means trained on ADNI directly using the same hyperparameters from MADRC. The ADNI data contains 192 time samples with a TR $= 3.0s$, whereas MADRC data contains 570 time samples with a TR $= 0.8s$. Therefore, this section explores the effect of temporal resolution on the predictive accuracy of MoCA and other key differences between the two data distributions. Similar to Sec. 7.5.2.1, Table 7.6 shows MoCA R values for the ADNI data under the zero-shot and all-shot settings.

FCM appears unaffected in both scenarios, achieving correlation values of $R = 0.12$, likely because it is independent of temporal dynamics. Notably, the CPM positive edge method showed a higher correlation value $R = 0.23$ on all-shot in contrast to zero-shot $R = 0.07$, indicating some distribution shift between MADRC and ADNI "brain scores". The model-based timeseries methods did not exhibit strong zero-shot transfer or all-shot performance, with G-ICA being the

Method	Zero-shot	All-shot
FCM + KRR	0.12^{***}	0.12^{***}
CPM_{pos} + KRR	0.07^{*}	0.23^{***}
CPM_{neg} + KRR	0.05^{*}	0.18^{***}
I-ICA + KRR	0.04	0.05^{*}
G-ICA + KRR	0.03	0.29^{***}
ALFF + KRR	0.05^{*}	0.01
TCN	0.09^{**}	0.16^{***}
BiLSTM	0.12^{***}	0.29^{***}
PatchTST	0.08^{**}	0.08^{**}
NeuroMamba	0.17^{***}	0.36^{***}

Table 7.6: Out of domain (OOD) generalization on Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset where values indicate MoCA Pearson correlation. Asterisks denote statistical significance as follows: $* = p < 0.1$, $** = p < 0.01$, $*** = p < 0.001$.

NeuroMamba	MoCA
Zero-shot	0.17^{***}
Single-shot	0.22^{***}
Three-shot	0.30^{***}
Five-shot	0.35^{***}
All-shot	0.36^{***}

Table 7.7: Domain adaptation on Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset where values indicate MoCA Pearson correlation. Asterisks denote statistical significance as follows: $* = p < 0.1$, $** = p < 0.01$, $*** = p < 0.001$.

notable exception, indicating that the ideal hyperparameters are a function of temporal resolution. Switching to data-driven timeseries methods, PatchTST and TCN showed weak zero-shot and all-shot performance, likely because the ideal patch size depends on the temporal resolution and kernel sizes. The BiLSTM model performed relatively well compared to the PatchTST and TCN methods, with an all-shot score of $R = 0.29$. The NeuroMamba model exhibited the strongest zero-shot transfer among all methods, with $R = 0.17$, and showed the most robustness to temporal sampling effects, matching literature observations comparing SSMs to other models [228]. When trained directly on ADNI, NeuroMamba achieved the highest all-shot performance ($R = 0.36$), indicating strong generalizability to other datasets.

7.5.2.9 Domain Adaptation

For NeuroMamba, per the results in Sec. 7.5.2.8, there is a gap between zero-shot $R = 0.17$ and all-shot $R = 0.36$ values. This provides an opportunity to study whether it is possible to close the gap by learning on a tiny subset of ADNI data, using the pretrained MADRC model to adjust

for any distribution mismatch. In the literature, there are many ways to do this, such as finetuning, freezing the backbone and updating only the last few layers, or using deep learning approaches such as correlation alignment (CORAL) [264], designed to minimize covariance differences between domains. In this setting, we found great success in simply finetuning the pretrained model. As observed in Table 7.7, with only 5 subjects per class for training data, NeuroMamba achieves an $R = 0.35$ value, nearly matching all-shot performance. This finding demonstrates the model's ability to adapt to a different domain with very limited training data, which is highly practical for many fMRI-related applications.

7.6 Conclusion

In this chapter, we sought to advance understanding and prediction of cognitive performance in AD by leveraging rs-fMRI data alongside behavioral scores, including MoCA, average memory, and average language metrics. By systematically evaluating the predictive power of both functional connectivity and multivariate timeseries data, we addressed limitations in prior studies that focused exclusively on functional connectivity. Our deep learning approach, based on state space modeling, demonstrated superior performance in predicting behavioral metrics, underscoring the value of temporal dynamics present in rs-fMRI data. Furthermore, NeuroMamba achieved a remarkable few-shot transfer performance on ADNI data, indicating that only a few subjects are needed when finetuning the MADRC-pretrained model on out-of-domain datasets such as ADNI, which is important for real-world usability. These results highlight the potential to integrate machine learning with neuroimaging to support early intervention for cognitive decline, using techniques such as HD-tDCS, while simultaneously offering deeper biological insights into AD progression.

However, this proposed approach and methodology have limitations. The R values across the behavior scores are modest, indicating possible limitations with using *resting-state* fMRI to analyze cognition scores. Perhaps these results are highly dependent on the parcellation scheme used. Additionally, it is likely that *task-based* fMRI, in which a patient performs a task under the scanner, could better "stress" brain networks, revealing a stronger relationship between key regions and behavioral metrics. For future work, one can explore face-name association [265] and object-location association [266] tasks within a *task-based* functional MRI framework to compare with the findings in this work and gain further insights by systematically contrasting resting-state and task-based functional activity.

7.7 Preliminary Results on Task-Based Functional MRI

7.7.1 Introduction

Task-based fMRI involves engaging patients in specific cognitive tasks during scanning, allowing researchers to observe how AD pathology impacts brain regions associated with functions such as memory, attention, or language. Prior studies have focused on resting-state fMRI, mainly due to abundant data from the Alzheimer's Disease Neuroimaging Initiative (ADNI), which lacks task-based fMRI since it is not included in the MRI protocol. Therefore, task-based fMRI as a modality for behavior score prediction is a relatively underexplored area that warrants investigation, since it may do a better job at "stressing" the brain through cognitive workloads. In this section, we use face-name association [265] and object-location association [266] task data, and present connectivity-based and model-based time series feature extraction methods that directly leverage the blood-oxygenation-level-dependent (BOLD) time series to learn a sparse collection of brain regions predictive of behavior scores.

In contrast to resting-state, task-based fMRI (tb-fMRI) involves engaging patients in specific cognitive tasks during scanning, such as finger tapping to activate regions in the motor cortex. Generally, it is known that task-based BOLD data can emphasize brain-behavior relationships [267, 266, 268]. Thus, one of the main topics of this section is to determine whether task-based fMRI protocols focused on memory and attention can stress relevant brain regions and networks to better capture relationships compared to resting-state fMRI. The resting-state fMRI scans were completed with eyes open with a fixation cross. Only a single baseline run is done for each subject for resting-state data. In this section, the face-name [265] and object-location [266] association fMRI tasks were performed as published. The face-name and object-location association tasks used a mixed event block design that consists of 6 active blocks and 7 rest blocks. Active blocks contain 3 novel and 3 repeated stimuli. Namely, for face-name runs, participants are shown faces and need to recall the associated name during the cued recall phase. Afterwards, they select the correct name from three options in the recognition phase. For object-location runs, participants are shown an object, and in the free recall phase, they must touch its location on a touch screen. In the cued recall phase, they see the object and its room (without the object) and must indicate the correct location. Later, in the recognition phase, they choose the object's location from three possible choices. The face-name data consists of two runs, while the object-location data is a single run only.

7.7.2 Related Works

To the best of the authors' knowledge, only one work has studied MoCA score prediction in a task-based fMRI context for AD, but strictly for MCI subjects [176, Chapter 2]. However, this work only considers CPM and PLS-BETA [269] methods that operate by using functional connectivity only, with the best results showing a Pearson correlation value of 0.04 for face-name and 0.22 for object-location tasks, and both of these were found to not be statistically significant ($p > 0.05$). It is worth noting that this is not a completely fair comparison since these works use different fMRI datasets and methodologies. However, these works only consider functional connectivity data, which collapses the time dimension and instead only looks at brain region interactions. To the best of the authors' knowledge, there are no task-based fMRI works that tackle behavior score prediction on the BOLD timeseries data, which may provide a richer context for finding more optimal predictions.

7.7.3 Setup

Due to the limited size of our dataset and the need to ensure generalization across the full distribution, we adopt a leave-one-out approach. Specifically, for each fold, the methods are trained on all data except one sample, which is held out for testing. This practice is commonly used in related fMRI prediction studies [176, 200, 198]. The {FCM, CPM, I-ICA, G-ICA, ALFF, fALFF} algorithms perform feature extraction by optimizing a convex cost function, and subsequently employ kernel ridge regression (KRR), which is also a convex optimization. Consequently, these methods are run until the optimization process converges. All relevant hyperparameters for the various methods were determined via cross-validation on resting-state fMRI data. To promote fair assessment of model generalizability, these parameters remain fixed across all folds and are not re-optimized for each individual fold. Afterwards, the methods were trained on face-name and object-location association datasets using those resting-state optimal hyperparameters in a leave-one-out setting.

7.7.4 Experiments

7.7.4.1 Predictive Accuracy

Table 7.8 contains Pearson correlation values for face-name and object-location association task data across the mentioned connectivity and timeseries-based methods. Further, the effects of the CONN Toolbox processing are explored by providing correlation values before and after this processing step. Focusing on MoCA specifically, the CPM method performed the best with correlation values of 0.35 and 0.37 for face-name and object-location tasks. These values are not substantially

higher than the NeuroMamba model in resting-state data, which has a correlation value of 0.36. For memory and language metrics, the highest values occurred on CPM using object-location data, with memory being 0.46 and language being 0.35. These values are higher than NeuroMamba on resting-state data, with values being around 0.25 for memory and language.

Generally, for the connectivity-based methods, CONN Toolbox processing proved advantageous for predictive accuracy. However, for the timeseries methods, CONN Toolbox processing was more of a mixed bag. It helped I-ICA and G-ICA methods, but hurt ALFF performance. It is not entirely clear why this processing helps some methods while hurting others. The only claim being made is that careful consideration of preprocessing steps is necessary, as it can affect performance in functional MRI analysis.

For all of the deep learning methods mentioned in this chapter and included in Table 7.3, these reported lower correlation values in the task-based regime compared to the resting-state regime. This occurred regardless if the model was pretrained on resting-state data and finetuned on task-based data, trained jointly with resting-state and task-based data, or only trained on task-based data. The same event occurred with and without CONN Toolbox processing, and across all behavior scores. Likely, there is not enough task-based data to do well on this problem. However, many approaches can be explored to help mitigate performance issues in this task-based domain. Future work can focus on deep learning techniques such as data augmentation or domain adaptation to increase predictive power in the task regime.

7.7.4.2 Important Regions

Given the results in Table 7.8, the CPM method performed very well across both face-name and object-location association tasks. Given the popularity of this method in functional MRI settings, it would be interesting to learn more about which nodes were useful to compute the "brain score" for each subject. Since a binary mask is constructed from the functional connectivity data, one can sum the number of nonzero entries on the columns to derive the degree of each node. Thus, one can plot the top 10 important nodes that were useful for behavior score prediction. Focusing on MoCA specifically, Table 7.9 contains the top 10 nodes for the task-based regime.

From this, it seems many regions from the sensory hand motor group were useful for MoCA prediction. This is not surprising since a subject has to answer multiple-choice questions with their hand. It is possible that a subject who is unsure about a face-name question tends to move their hand more often, going back and forth between multiple options. Therefore, in a way, one is predicting MoCA performance using information about the exam taken under the scanner. This score-to-score prediction is interesting, but not the scope of the project. We are more interested in predicting MoCA scores through neural activity, so future work can focus on using a subset of Power atlas ROIs that do not include the sensory motor group regions. However, the most

Method	Before CONN [205] MoCA	Memory	Language	After CONN [205] MoCA	Memory	Language
			Face-Name [265]			
FC	0.17^{***}	0.09^{*}	0.16^{**}	0.30^{***}	0.27^{***}	0.23^{***}
CPM^{+}	0.13^{**}	0.12^{**}	0.16^{**}	0.35^{***}	0.16^{**}	0.20^{***}
CPM^{-}	0.12^{**}	0.13^{**}	0.13^{**}	-0.03	0.05	0.03
I-ICA	-0.06	-0.09^{*}	0.09^{*}	0.07	-0.05	0.06
G-ICA	0.24^{***}	0.08	0.16^{**}	0.26^{***}	0.18^{***}	0.26^{***}
ALFF	0.37^{***}	0.29^{***}	0.34^{***}	0.18^{***}	0.18^{***}	0.24^{***}
fALFF	0.27^{***}	0.22^{***}	0.26^{***}	N/A	N/A	N/A
			Object-Location [266]			
FC	-0.01	-0.02	0.03	0.37^{***}	0.36^{***}	0.25^{*}
CPM^{+}	0.34^{***}	0.39^{***}	0.05	0.34^{***}	0.46^{***}	0.35^{***}
CPM^{-}	-0.12	0.11	-0.13	0.09	0.16	0.12^{***}
I-ICA	0.04	-0.12	0.16	0.07	-0.10	-0.18^{*}
G-ICA	-0.02	0.19^{*}	0.04	0.21^{*}	0.27^{**}	0.20^{*}
ALFF	0.25^{*}	0.25^{*}	0.31^{**}	0.11	0.13	0.33^{**}
fALFF	0.28^{**}	0.25^{*}	0.30^{**}	N/A	N/A	N/A

Table 7.8: Pearson correlation coefficients (R) and corresponding p-values (p) by score category for multiple methods applied to the MADRC rs-fMRI data. Asterisks denote statistical significance as follows: $* = p < 0.1$, $** = p < 0.01$, $*** = p < 0.001$.

Node Degree	Power ROI	MNI (x,y,z)	Nominal System	Lobe/Area	Talairach Daemon Label
Face-Name Association [265]					
70	272	(-29, -38, -4)	Memory retrieval	Temporal Lobe	Sub-Gyral
67	268	(30, -24, -12)	Memory retrieval	Sub-lobar	Extra-Nuclear
64	43	(36, -9, 14)	Sensory/somatomotor Mouth	Sub-lobar	Insula
56	215	(0, 30, 27)	Salience	*	*
51	15	(0, -15, 47)	Sensory/somatomotor Hand	Frontal Lobe	Paracentral Lobule
51	271	(-30, -24, -12)	Memory retrieval	Sub-lobar	Extra-Nuclear
50	29	(44, -8, 57)	Sensory/somatomotor Hand	Frontal Lobe	Precentral Gyrus
43	203	(11, -39, 50)	Salience	Parietal Lobe	Precuneus
42	131	(-49, -42, 1)	Default mode	Temporal Lobe	Sub-Gyral
42	252	(-52, -63, 5)	Dorsal attention	Temporal Lobe	Middle Temporal Gyrus
Object-Location Association [266]					
60	93	(15, -63, 26)	Default mode	Occipital Lobe	Precuneus
54	152	(-18, -68, 5)	Visual	Occipital Lobe	Cuneus
43	20	(-54, -23, 43)	Sensory/somatomotor Hand	Parietal Lobe	Postcentral Gyrus
42	26	(50, -20, 42)	Sensory/somatomotor Hand	Frontal Lobe	Postcentral Gyrus
41	33	(-45, -32, 47)	Sensory/somatomotor Hand	Parietal Lobe	Inferior Parietal Lobule
41	61	(32, -26, 13)	Auditory	Sub-lobar	Insula
40	155	(-14, -91, 31)	Visual	Occipital Lobe	Cuneus
39	29	(44, -8, 57)	Sensory/somatomotor Hand	Frontal Lobe	Precentral Gyrus
39	57	(-34, 3, 4)	Cingulo-opercular Task Control	Sub-lobar	Claustrum
35	62	(65, -33, 20)	Auditory	Temporal Lobe	Superior Temporal Gyrus

Table 7.9: Top ten brain regions implicated in MoCA score prediction, ranked by node degree using CPM method. Additional columns provide MNI coordinates, nominal system category, lobe classification, and Talairach Daemon (TD) labels.

important nodes selected across both tasks include regions associated with memory retrieval, the default mode network, and visual processing areas. This indicates that there is likely neural activity in these regions that appears to correlate with MoCA prediction.

7.7.4.3 Brain Network Visualizations

Given the results in Table 7.8, the CPM method performed very well across both face-name and object-location association tasks. Given the popularity of this method in functional MRI settings, it would be interesting to learn more about the binary mask used to compute the "brain score" for each subject. By loading the mask into Yale BioImage Suite Connectivity Viewer (bioimagesuiteweb.github.io/webapp/connviewer.html), one can visualize the most important edge connections between nodes. For the reported figures, an arbitrary node threshold is set at 50 to prevent the diagram from appearing overly busy. In Fig. 7.7, the CPM binary mask for MoCA prediction on face-name data is shown in the left, followed by the connectivity diagram on the right with included network definitions. In Fig. 7.8, the CPM results for MoCA prediction on object-location data are shown. For both figures, the positive edge variant of CPM is used, given the higher

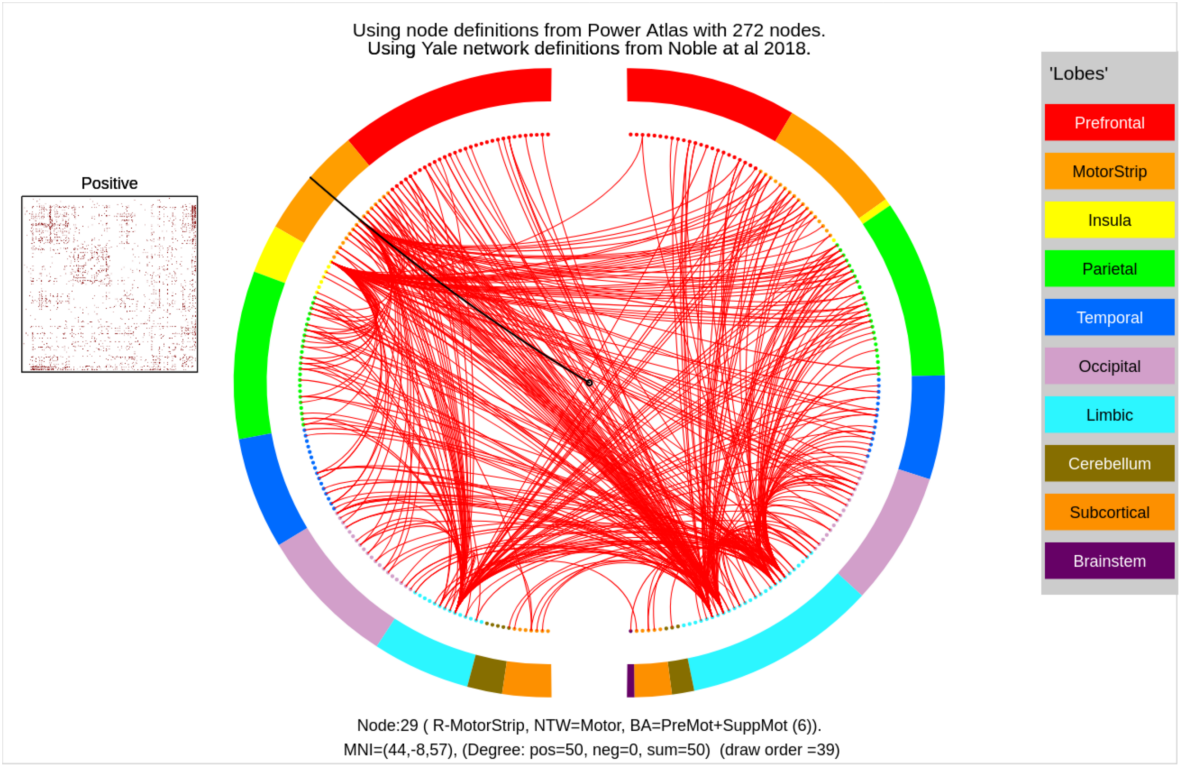


Figure 7.7: Yale BioImage Connectivity Viewer on face-name association task.

predictive accuracy when compared to the negative edge variant of CPM. From these results, it appears that connectivity values related to the Limbic cortex and the Occipital lobe were important for MoCA prediction across both association tasks. However, as mentioned in Sec. 7.7.4.1, various connections to the motor strip are included in these diagrams. Thus, it would be interesting to remove sensory motor regions and visualize the sub-networks that relate to neural activity rather than task activity.

7.7.4.4 Conclusion

This section explored task-based fMRI for behavioral score prediction using face-name and object-location association tasks. The CPM method produced the strongest and most consistent results, with moderate correlations but not substantially higher than values observed in resting-state fMRI using the NeuroMamba model. CONN Toolbox preprocessing improved connectivity-based prediction but had mixed effects for time-series feature methods, underscoring that preprocessing choices can substantially change outcomes. Interpretation of CPM masks suggested involvement of memory, default-mode, and visual regions, but also highlighted strong contributions from sensory motor hand areas, indicating potential issues with score-to-score prediction as opposed to

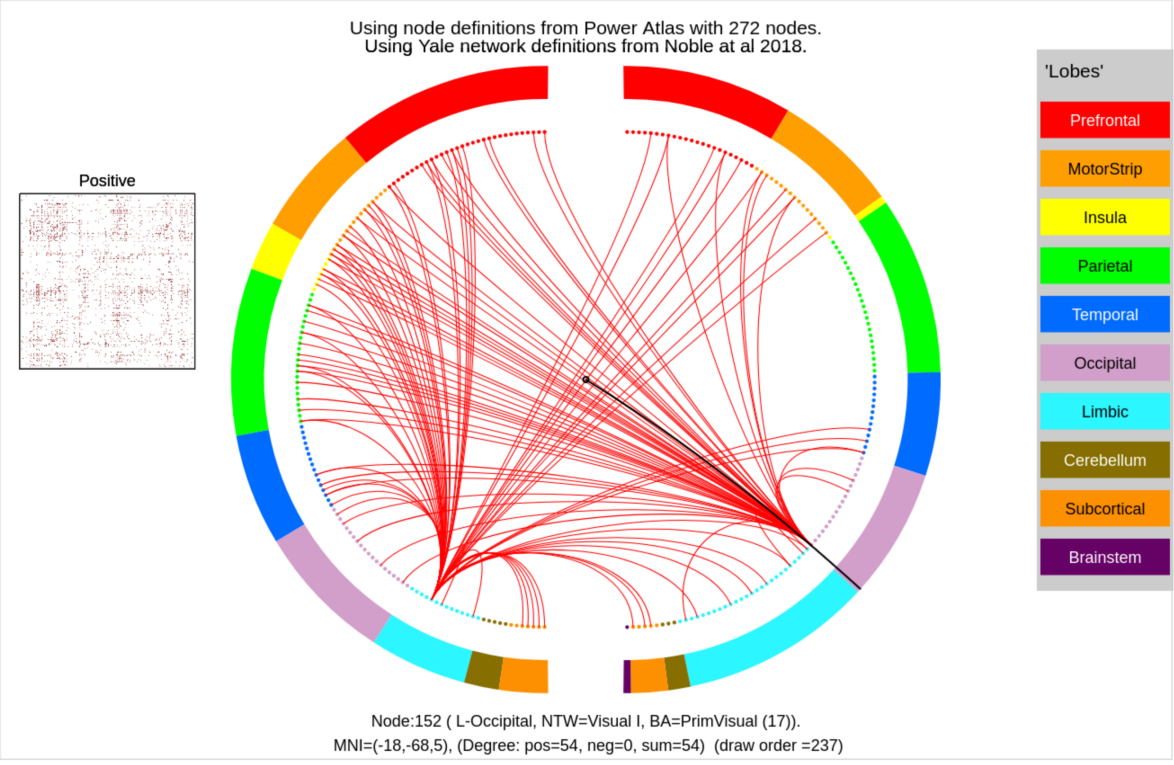


Figure 7.8: Yale BioImage Connectivity Viewer on object-location association task.

what we want, that being, BOLD-to-score prediction. Additionally, deep learning models underperformed in the task-based setting across training strategies, likely due to limited sample size, or a possible incorrect hypothesis that task-based fMRI would lead to stronger correlations for MoCA prediction from the "brain stressing" task protocols found in face-name and object-location tasks.

7.7.4.5 Towards Publication

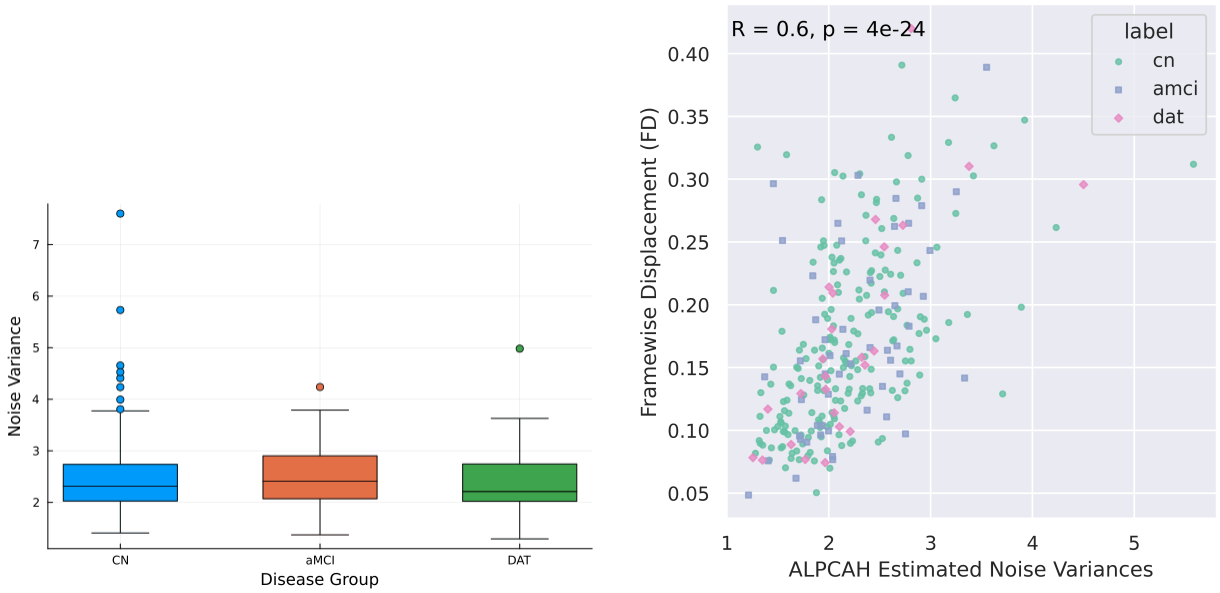
As of now, this work, based on preliminary results on task-based functional MRI and behavior score prediction, remains unpublished. A few key improvements and additions are necessary to get this work closer to publication. Importantly, the results reported in this section involve the z-score normalized metrics, but not those that were regressed for age and education. From Table 7.1, there seems to be a lot of overlap between age and education for all three disease categories, so there should not be drastic differences in prediction once adjusted. However, for good measure, the scores should have age and education regressed out to prevent the models from learning age or education that may possibly be inherent in the BOLD data. Secondly, the brain regions associated with the sensory motor group, like hands and mouth, should be removed from the BOLD data to prevent information leakage about the task done in the scanner, as mentioned earlier. Thirdly,

while the effects of CONN processing are explored for task-based data, this is missing for resting-state data. For completeness, this publication should include results on resting-state data with and without CONN Toolbox processing in Table 7.8. Lastly, it would be interesting to add a general linear model (GLM) and dynamic causal model (DCM) results to Table 7.8 since they rely on knowing the task and rest intervals, so we can measure the correlation between task activity and BOLD timeseries.

7.8 Additional Results

7.8.1 Heteroscedastic Motion Modeling with ALPCAH

Chapter 3 presented a heteroscedastic subspace modeling named LR-ALPCAH that can jointly estimate a subspace basis and the noise variances associated with each sample. Chapter 7 presented behavior score prediction in functional MRI. Since some subjects exhibit more motion than others, an interesting research problem is whether ALPCAH can be used to measure average motion for each subject as a form of heteroscedastic modeling. Instead of working with the raw timeseries data, the frequency magnitudes between 0.01-0.30Hz are extracted for each brain region and collected into a single vector for each subject. The frequency magnitudes are better suited for this motion modeling task because, between multiple subjects, they may show high motion at different time samples. In contrast, the frequency magnitude information would stay the same, assuming the physical movement is identical. Once the matrix of frequency magnitudes is formed, the matrix is demeaned based on the sample mean. Afterwards, the SignFlipPA [79] method is used to compute an estimate of the rank of the matrix, which gave a value of $\hat{d} = 5$. With this information at hand, LR-ALPCAH is used to compute noise variances for each subject. Interestingly, there are no substantial differences between noise variance distributions among the disease categories. One may expect that DAT subjects have higher motion due to Parkinson-like conditions or general symptoms from AD, but this does not seem to be the case for our data. However, using `fmriprep`'s frame-wise displacement (FD) values, one can compare ALPCAH noise variances to these motion estimation values, called FD, that are computed during the preprocessing of our data. Fig. 7.9 illustrates our findings, and more importantly, shows a correlation value of $R = 0.6$ between ALPCAH estimated noise variances and FD values. This shows some utility in heteroscedastic subspace modeling for motion estimation and correction applications in functional MRI. Future work can explore this relationship further, potentially illustrating comparisons between motion-corrected BOLD timeseries using heteroscedastic modeling and raw BOLD timeseries data.



(a) ALPCAH estimated subject noise variances broken up by disease category.

(b) Correlation plot between frame-wise displacement (FD) and ALPCAH noise variances.

Figure 7.9: Heteroscedastic subspace learning as a motion estimation/correction model in fMRI.

CHAPTER 8

Future Work

8.1 ALPCAH: Subspace Learning for Heteroscedastic Data

Chapter 3 explored a PCA-like algorithm that can simultaneously estimate a subspace basis while estimating data quality via noise variances. However, many extensions and scenarios not considered in this work would be interesting to explore.

- **Doubly Heteroscedastic Models**: In this chapter, heteroscedasticity is only considered across data samples, but it is also possible that the feature space is heteroscedastic. For example, if air quality data is collected over time, the sensors may become less reliable, requiring recalibration (sample space), while also having sensors of different quality, ranging from military-grade to commercial products (feature space). There has been work in this area, such as the Dyson equalizer algorithm [270] that equalizes the noise variance across both rows and columns via a normalization technique that works on a broad range of noise distributions. Interested readers should check out this work before exploring this subtopic.

- **Deep Heteroscedastic Models:** In this chapter, only subspace models are considered, which are elegantly simple. The reality is that nonlinear models achieve state-of-the-art performance in many machine learning applications, thanks to the rise of deep learning and neural networks. For a network trained to do classification/regression tasks, does the model inherently learn which data samples are more reliable? If not, is there a way to simultaneously perform the task of interest while estimating data quality and use this information to improve task performance? These are the questions that would be fascinating to explore.

- **Manifold Heteroscedastic Models:** Similarly to above, this chapter explored subspace learning, but for many datasets, the data is more complicated and requires more sophisticated modeling. There are many learning algorithms, such as Multidimensional scaling (MDS), t-SNE, and UMAP, that also find a lower-dimensional space on manifolds. Given

that MDS is the same as PCA when using the Euclidean norm as the distance metric, is there a way to adapt ALPCAH using some other metric? This topic is possibly less interesting because many manifold learning algorithms exist for visualization rather than analysis, so its utility may be limited.

- **Heteroscedastic Regression Models:** Linear regression and subspace learning are very similar tasks, with the subtle point that the residual error goes from subspace projection $\|\boldsymbol{y} - \boldsymbol{U}\boldsymbol{U}^T\boldsymbol{y}\|$ to regression error $\|\boldsymbol{y} - A\boldsymbol{x}\|$. Contrary to the name, linear regression just means linear with respect to the coefficients. Nonlinear models, such as quadratic equations and more general polynomials, can be learned with linear regression. From preliminary searches, there are a few papers already that talk about estimating heteroscedasticity in linear regression. However, there might be interesting, unexplored directions, such as nonlinear regression via MLPs in heteroscedastic settings.

- **General Noise Distribution Models:** In many real-world settings, Gaussian noise is a fair assumption that holds in practice, even when not exactly true. However, there are many applications, such as SPECT imaging, phase retrieval, and semiconductor metrology analysis, where the noise distribution is known to follow a Poisson distribution. In this chapter, although the subspace basis coordinates do not follow a Gaussian distribution, we assume the noise is Gaussian. Therefore, generalizations of ALPCAH for Poisson and other distributions could be fruitful if there are sufficient applications with such distributions.

- **Heteroscedastic Tensor Models:** Subspace models assume a matrix of N data samples $\boldsymbol{Y} \in \mathbb{R}^{D\times N}$ of D-dimensional space. This may be too limiting for non-vector data such as functional MRI that is 4D due to space and time $\boldsymbol{Y} \in \mathbb{R}^{X\times Y\times Z\times T}$. Reshaping this into a matrix would lose important spatial relationships, since nearby brain voxels exhibit similar BOLD activation patterns. Generalizing ALPCAH for tensor objects would be another direction of future work. For example, 3D CT data from low-dose and high-dose samples could be seen as heteroscedastic across samples. Perhaps tensor learning would enhance image quality for low-dose samples better than a union of subspace or dictionary learning approaches.

- **Outlier and Heteroscedastic Hybrid Models:** This work considered Robust PCA as a comparison method to LR-ALPCAH, which is focused on heteroscedastic noise specifically. Future work can consider a hybrid model that adds a matrix to capture outliers in a more general setting. One possible formulation would be $\frac{1}{2}\|(\boldsymbol{Y} - \boldsymbol{L}\boldsymbol{R}' - \boldsymbol{E})\boldsymbol{\Pi}^{-1/2}\|_{\mathrm{F}}^2 + \frac{D}{2}\log|\boldsymbol{\Pi}| + \lambda\|\boldsymbol{E}\|_{1,1}$ where we note that taking $\lambda \to \infty$ leads to the standard LR-ALPCAH formulation.

It is unclear if there would be an advantage to adding an outlier model when the outlier can be treated as a very noisy (heteroscedastic) sample, but experimentation can prove differently.

8.2 ALPCAHUS: Subspace Clustering for Heteroscedastic Data

Chapter 4 explored a subspace clustering algorithm that can simultaneously estimate multiple subspace bases while estimating data quality via noise variances. However, many extensions and scenarios not considered in this work would be interesting to explore.

- **Most topics above from subspace learning:** Since subspace clustering is a union of subspace models, there are similar things to explore, such as general noise distribution models, doubly heteroscedastic models, tensor models, and deep clustering models, to name a few.

- **Heteroscedastic Dictionary Learning:** There are many similarities between subspace clustering and dictionary learning, with the distinction that the dictionary is often more general than a subspace model. Dictionary learning will solve for dictionary $\boldsymbol{D}$ and atoms $\boldsymbol{Z}$ with the following

$$\min_{\boldsymbol{Z},\boldsymbol{D}} \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{Z}\|_F^2 + \mathcal{R}(\boldsymbol{Z}) \tag{8.1}$$

  where $\mathcal{R}(\cdot)$ is a regularizer such as $\|\cdot\|_{1,1}$. If you constrain $\boldsymbol{D} = \boldsymbol{Y}$, then dictionary learning almost becomes Sparse Subspace Clustering (SSC). Hence, there are interesting directions to explore that might prove useful since a union of subspace models might be too limiting for some datasets, especially if the data is heterogeneous. Heteroscedastic modeling in dictionary learning could be useful for medical applications, such as low-dose and high-dose CT imaging, where some image patches are noisier than others due to lower radiation doses.

- **Beyond Subspaces:** Similar to dictionary learning, a subspace model can be limiting for some datasets. The kernel trick has been a useful tool for developing methods such as Kernel PCA and Kernel Support Vector Machines (SVMs). ALPCAHUS could be generalized using Gaussian kernels for datasets that do not follow a union-of-subspaces model, increasing its utility to general clustering problems.

8.3 PET-TURTLE: Deep Unsupervised Support Vector Machines for Imbalanced Data Clusters

In Chapter 5, a deep clustering algorithm is developed to better handle imbalanced datasets using the Support Vector Machine (SVM) formulation. It is based on the TURTLE algorithm, which is meant for foundational model latent variables only. It alternates between finding the max-margin hyperplane and the classifier, like K-means. Some extensions not considered in this work would be interesting to explore.

- **Joint Encoding & Clustering:** Since TURTLE requires latent variables extracted from a foundational model, it means that one must use a pretrained CLIP-like or contrastive-like model. However, if the data falls outside the realm of 2D images/text, one is forced to train a representation-learning model from scratch. For example, 3D structural MRI data would require training. Because of this, it may be useful to jointly encode and cluster data to improve results, including finetuning pretrained models. This means having a cost function that combines the contrastive learning equation (pulling similar samples together in latent space while repelling dissimilar samples) and the TURTLE clustering function (cross-entropy loss) to promote a well-separated embedded space. Through joint estimation, one would hope that the clustering accuracy is closer to a linear probe applied to a contrastive model.

- **Kernel SVMs:** The TURTLE formulation finds the max-margin hyperplane, which assumes linear separability of the input data. This is fine since the input data comes from foundational or representation-learning models trained in a CLIP or contrastive fashion. However, this method will not generalize well to input data that is not linearly separable, for example, when working with raw data rather than latent data. Thus, there is an opportunity to use Radial Basis Functions (RBFs) to find a more suitable nonlinear hyperplane that better separates data clusters. Doing this may be more complicated than expected, since the theory says that gradient descent with the cross-entropy loss will produce iterates biased towards the direction of the max-margin hyperplane. Modifying the cost function carelessly could break this important relation, so additional work is required to mesh both ideas together.

8.4 Alzheimer's Disease Diagnosis in Functional MRI via 4D Convolutions

In Chapter 6, a 4D classification model is developed for functional MRI. Many things not considered here would be interesting to explore.

- **Gray & White Matter Modalities:** In our work, we focused on functional MRI, which captures BOLD activity in gray matter regions. There is interesting and new work by Amaya Murgia and Andrea Jacobson in the fMRI lab, on using myelin water imaging as a biomarker for Alzheimer's disease. This modality examines white matter regions that have important implications for understanding brain plasticity, as they determine the speed of action potentials. See [271] for an overview of this topic. Thus, using fMRI and myelin imaging together for diagnosis could prove extremely useful since both gray and white matter data complement each other in Alzheimer's disease applications.

- **Motion Correction:** The datasets used in this project contain functional MRI data from older patients. Because of this, time samples sometimes exhibit heavy motion. In the fMRI preprocessing pipeline, one can determine which time samples experience motion by inspecting the "FD" vector from fmriprep. An interesting direction of work could be to treat this as missing/masked timeseries data and attempt to impute the values. Ideally, the model would be robust to both task and resting-state data. As shown in Sec. 7.8.1, there was some advantage to heteroscedastic subspace learning for motion modeling and correction, so this would be an interesting direction that merges heteroscedastic learning and functional MRI.

8.5 Behavior Score Prediction in Resting-State and Task-Based Functional MRI

In Chapter 7, a deep learning model for predicting behavior scores in Alzheimer's disease is proposed for resting-state functional MRI. There are a few directions worth exploring.

- **Task-based Functional MRI:** Naturally, an open question remains whether task-based functional MRI, such as face-name association or object-location association tasks, can better help uncover brain-behavior relationships and show stronger correlation between behavior score metrics such as MoCA and functional activity. Some preliminary work in this direction was shown in the chapter. More steps are necessary for publication; refer to Sec. 7.7 for more information.

- **Alzheimer's Disease Progression (Longitudinal Study):** In Fig. 7.5, it was shown that using MoCA in conjunction with rs-fMRI did not produce a meaningful increase in ROC characteristics and AUC values in our dataset with our specific preprocessing pipeline and methods. However, it would be interesting to see if one can identify which healthy subjects are likely to become amnestic MCI (aMCI) or dementia of the Alzheimer's type (DAT) using rs-fMRI. This is challenging since only baseline scans exist for these subjects in this current

timeframe. However, it is possible that, in the future, MADRC will have a small, usable dataset of subjects who progressed to a different disease category. Predicting cognitive decline and time to conversion given fMRI data would be interesting, as one usually does not have fMRI data for the same subject when healthy and non-healthy.

- **Multiple Biomarkers:** MADRC has collected blood biomarkers that measure tau protein levels. An in-depth analysis of diagnosis performance and disease progression using blood biomarkers, behavior scores such as MoCA, and genetic/family information could be a potential project to investigate whether blood biomarkers provide complementary information. Determining whether each provides a distinct advantage in diagnosis, rather than redundancy, among these readily available biomarkers would be a potential future topic. Likely, only classical ML methods would be useful due to limited data, so the innovation would be on the medical and biomedical implications.

- **Linking fMRI and Tau Proteins:** Instead of predicting behavior scores, an alternative would be to explore whether tau protein buildup, via blood biomarkers as mentioned above, has a stronger correlation than MoCA or similar metrics. A model could predict a spatial map of the brain indicating the level of tau build-up in each region, whose sum of the brain volume corresponds to the overall level of tau protein in the blood.

- **Sensitive Feature Removal:** One discussion topic raised from this project revolved around the possibility of removing age, education, sex, and race features from the functional MRI data. This is a challenging question when working with the 4D spatial-temporal data or timeseries data. For metrics, this is quite easy to do using linear regression and extracting the residuals to be the new metrics. However, for high-dimensional functional MRI data, there is no clear way to regress the features out of it. Thus, there is an open question whether some method could be applied that works regardless of the input data. One could train a deep model to predict age or education, assess how correlated the predicted and true values are, and, from this, remove them in some general way. For example, in image classification, if input X is classified as being a panda, could one create $X + \epsilon$ such that it is no longer a panda using that trained model? This is an ill-posed problem, since "not a panda" is very ambiguous. See adversarial networks that add slight noise to the input X to change the class, yet the image still visually looks like a panda. A challenging problem with many possible open research directions.

BIBLIOGRAPHY

[1] M. N. Moussa, M. R. Steen, P. J. Laurienti, and S. Hayasaka, "Consistency of network modules in resting-state fMRI connectome data," *PLOS ONE*, vol. 7, no. 8, p. e44428, 2012.

[2] S. G. Mueller, M. W. Weiner, L. J. Thal, R. C. Petersen, C. Jack, W. Jagust, J. Q. Trojanowski, A. W. Toga, and L. Beckett, "The Alzheimer's disease neuroimaging initiative," *Neuroimaging Clinics of North America*, vol. 15, no. 4, p. 869, 2005.

[3] L. Fischer, E. N. Molloy, A. P. Binette, N. Vockert, J. Marquardt, A. P. Pilar, M. C. Kreissl, J. Remz, J. Tremblay-Mercier, J. Poirier, *et al.*, "Precuneus activity during retrieval is positively associated with amyloid burden in cognitively normal older APOE4 carriers," *Journal of Neuroscience*, vol. 45, no. 6, 2025.

[4] Z. S. Nasreddine, N. A. Phillips, V. Bédirian, S. Charbonneau, V. Whitehead, I. Collin, J. L. Cummings, and H. Chertkow, "The montreal cognitive assessment, MoCA: a brief screening tool for mild cognitive impairment," *Journal of the American Geriatrics Society*, vol. 53, no. 4, pp. 695–699, 2005.

[5] J. Bennett and S. Lanning, "The netflix prize," 2007.

[6] Y. Koren, "The BellKor solution to the netflix grand prize," *Netflix prize documentation*, vol. 81, no. 2009, pp. 1–10, 2009.

[7] M. D. Greicius, B. Krasnow, A. L. Reiss, and V. Menon, "Functional connectivity in the resting brain: a network analysis of the default mode hypothesis," *Proceedings of the national academy of sciences*, vol. 100, no. 1, pp. 253–258, 2003.

[8] M. E. Raichle, A. M. MacLeod, A. Z. Snyder, W. J. Powers, D. A. Gusnard, and G. L. Shulman, "A default mode of brain function," *Proceedings of the national academy of sciences*, vol. 98, no. 2, pp. 676–682, 2001.

[9] B. Biswal, F. Zerrin Yetkin, V. M. Haughton, and J. S. Hyde, "Functional connectivity in the motor cortex of resting human brain using echo-planar MRI," *Magnetic resonance in medicine*, vol. 34, no. 4, pp. 537–541, 1995.

[10] M. Hampson, B. S. Peterson, P. Skudlarski, J. C. Gatenby, and J. C. Gore, "Detection of functional connectivity using temporal correlations in MR images," *Human brain mapping*, vol. 15, no. 4, pp. 247–262, 2002.

[11] J. Xiong, L. M. Parsons, J.-H. Gao, and P. T. Fox, "Interregional connectivity to primary motor cortex revealed using MRI resting state images," *Human brain mapping*, vol. 8, no. 2-3, pp. 151–156, 1999.

[12] J. Salazar Cavazos, J. A. Fessler, and L. Balzano, "ALPCAH: Sample-wise heteroscedastic PCA with tail singular value regularization," in *2023 International Conference on Sampling Theory and Applications (SampTA)*, pp. 1–6, SampTA, 2023.

[13] J. Salazar Cavazos, J. A. Fessler, and L. Balzano, "ALPCAH: Subspace learning for sample-wise heteroscedastic data," *Transactions on Signal Processing (TSP)*, pp. 1–12, 2025.

[14] J. Salazar Cavazos, J. A. Fessler, and L. Balzano, "ALPCAHUS: Subspace clustering for heteroscedastic data," *arXiv preprint arXiv:2505.18918*, 2025.

[15] J. Salazar Cavazos, "PET-TURTLE: Deep unsupervised support vector machines for imbalanced data clusters," *IEEE Signal Processing Letters*, vol. 33, pp. 91–95, 2026.

[16] J. Salazar Cavazos and S. Peltier, "Alzheimer's disease classification in functional MRI with 4D joint temporal-spatial kernels in novel 4D CNN model," in *International Society for Magnetic Resonance in Medicine*, p. 6929, ISMRM, 2025.

[17] J. Salazar Cavazos, M. Egan, K. Litinas, B. Hampstead, and S. Peltier, "Behavior score prediction in resting-state functional MRI by deep state space modeling," *arXiv preprint arXiv:2602.07131*, 2026.

[18] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[19] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.

[20] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," *Electronics*, vol. 9, no. 8, p. 1295, 2020.

[21] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: why and how you should (still) use DBSCAN," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017.

[22] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," *ACM Sigmod record*, vol. 28, no. 2, pp. 49–60, 1999.

[23] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[24] P. K. Agarwal and N. H. Mustafa, "K-means projective clustering," in *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 155–165, 2004.

[25] K. B. Rajan, J. Weuve, L. L. Barnes, E. A. McAninch, R. S. Wilson, and D. A. Evans, "Population estimate of people with clinical Alzheimer's disease and mild cognitive impairment in the united states (2020–2060)," *Alzheimer's & dementia*, vol. 17, no. 12, pp. 1966–1975, 2021.

[26] C. R. Jack Jr, D. A. Bennett, K. Blennow, M. C. Carrillo, B. Dunn, S. B. Haeberlein, D. M. Holtzman, W. Jagust, F. Jessen, J. Karlawish, *et al.*, "Nia-aa research framework: toward a biological definition of Alzheimer's disease," *Alzheimer's & dementia*, vol. 14, no. 4, pp. 535–562, 2018.

[27] G. D. Rabinovici, M. C. Carrillo, M. Forman, S. DeSanti, D. S. Miller, N. Kozauer, R. C. Petersen, C. Randolph, D. S. Knopman, E. E. Smith, *et al.*, "Multiple comorbid neuropathologies in the setting of Alzheimer's disease neuropathology and implications for drug development," *Alzheimer's & Dementia: Translational Research & Clinical Interventions*, vol. 3, no. 1, pp. 83–91, 2017.

[28] T. Ohm, H. Müller, H. Braak, and J. Bohl, "Close-meshed prevalence rates of different stages as a tool to uncover the rate of Alzheimer's disease-related neurofibrillary changes," *Neuroscience*, vol. 64, no. 1, pp. 209–217, 1995.

[29] P. Vemuri and C. R. Jack, "Role of structural MRI in Alzheimer's disease," *Alzheimer's research & therapy*, vol. 2, pp. 1–10, 2010.

[30] M. D. Greicius, G. Srivastava, A. L. Reiss, and V. Menon, "Default-mode network activity distinguishes Alzheimer's disease from healthy aging: evidence from functional MRI," *Proceedings of the National Academy of Sciences*, vol. 101, no. 13, pp. 4637–4642, 2004.

[31] R. D. Nebes, "Semantic memory in Alzheimer's disease.," *Psychological bulletin*, vol. 106, no. 3, p. 377, 1989.

[32] J. R. Hodges and K. Patterson, "Is semantic memory consistently impaired early in the course of Alzheimer's disease? neuroanatomical and diagnostic implications," *Neuropsychologia*, vol. 33, no. 4, pp. 441–459, 1995.

[33] C. R. Jack Jr, D. A. Bennett, K. Blennow, M. C. Carrillo, H. H. Feldman, G. B. Frisoni, H. Hampel, W. J. Jagust, K. A. Johnson, D. S. Knopman, *et al.*, "A/T/N: An unbiased descriptive classification scheme for Alzheimer disease biomarkers," *Neurology*, vol. 87, no. 5, pp. 539–547, 2016.

[34] C. R. Jack Jr, D. S. Knopman, S. D. Weigand, H. J. Wiste, P. Vemuri, V. Lowe, K. Kantarci, J. L. Gunter, M. L. Senjem, R. J. Ivnik, *et al.*, "An operational approach to national institute on aging–Alzheimer's association criteria for preclinical Alzheimer disease," *Annals of neurology*, vol. 71, no. 6, pp. 765–775, 2012.

[35] L. Mosconi, R. Mistur, R. Switalski, W. H. Tsui, L. Glodzik, Y. Li, E. Pirraglia, S. De Santi, B. Reisberg, T. Wisniewski, *et al.*, "FDG-PET changes in brain glucose metabolism from normal cognition to pathologically verified Alzheimer's disease," *European journal of nuclear medicine and molecular imaging*, vol. 36, pp. 811–822, 2009.

[36] M. De Leon, A. Convit, O. Wolf, C. Tarshish, S. DeSanti, H. Rusinek, W. Tsui, E. Kandil, A. Scherer, A. Roche, *et al.*, "Prediction of cognitive decline in normal elderly subjects with 2-[18f] fluoro-2-deoxy-d-glucose/positron-emission tomography (FDG/PET)," *Proceedings of the National Academy of Sciences*, vol. 98, no. 19, pp. 10966–10971, 2001.

[37] L. Mosconi, A. Pupi, and M. J. De Leon, "Brain glucose hypometabolism and oxidative stress in preclinical Alzheimer's disease," *Annals of the New York Academy of Sciences*, vol. 1147, no. 1, pp. 180–195, 2008.

[38] M. Tondelli, G. K. Wilcock, P. Nichelli, C. A. De Jager, M. Jenkinson, and G. Zamboni, "Structural MRI changes detectable up to ten years before clinical Alzheimer's disease," *Neurobiology of aging*, vol. 33, no. 4, pp. 825–e25, 2012.

[39] M. D. Ikonomovic, W. E. Klunk, E. E. Abrahamson, C. A. Mathis, J. C. Price, N. D. Tsopelas, B. J. Lopresti, S. Ziolko, W. Bi, W. R. Paljug, *et al.*, "Post-mortem correlates of in vivo PiB-PET amyloid imaging in a typical case of Alzheimer's disease," *Brain*, vol. 131, no. 6, pp. 1630–1645, 2008.

[40] J. Poirier, P. Bertrand, S. Kogan, S. Gauthier, J. Davignon, and D. Bouthillier, "Apolipoprotein e polymorphism and Alzheimer's disease," *The Lancet*, vol. 342, no. 8873, pp. 697–699, 1993.

[41] J.-C. Lambert, S. Heath, G. Even, D. Campion, K. Sleegers, M. Hiltunen, O. Combarros, D. Zelenika, M. J. Bullido, B. Tavernier, *et al.*, "Genome-wide association study identifies variants at clu and cr1 associated with Alzheimer's disease," *Nature genetics*, vol. 41, no. 10, pp. 1094–1099, 2009.

[42] S. Seshadri, A. L. Fitzpatrick, M. A. Ikram, A. L. DeStefano, V. Gudnason, M. Boada, J. C. Bis, A. V. Smith, M. M. Carrasquillo, J. C. Lambert, *et al.*, "Genome-wide analysis of genetic loci associated with Alzheimer disease," *Jama*, vol. 303, no. 18, pp. 1832–1840, 2010.

[43] C. Bellenguez, F. Küccükali, I. E. Jansen, L. Kleineidam, S. Moreno-Grau, N. Amin, A. C. Naj, R. Campos-Martin, B. Grenier-Boley, V. Andrade, *et al.*, "New insights into the genetic etiology of Alzheimer's disease and related dementias," *Nature genetics*, vol. 54, no. 4, pp. 412–436, 2022.

[44] F. Bai, C. Xie, D. R. Watson, Y. Shi, Y. Yuan, Y. Wang, C. Yue, Y. Teng, D. Wu, and Z. Zhang, "Aberrant hippocampal subregion networks associated with the classifications of aMCI subjects: a longitudinal resting-state study," *PloS one*, vol. 6, no. 12, p. e29288, 2011.

[45] F. Agosta, M. Pievani, C. Geroldi, M. Copetti, G. B. Frisoni, and M. Filippi, "Resting state fMRI in Alzheimer's disease: beyond the default mode network," *Neurobiology of aging*, vol. 33, no. 8, pp. 1564–1578, 2012.

[46] M. A. Binnewijzend, M. M. Schoonheim, E. Sanz-Arigita, A. M. Wink, W. M. van der Flier, N. Tolboom, S. M. Adriaanse, J. S. Damoiseaux, P. Scheltens, B. N. van Berckel, *et al.*, "Resting-state fMRI changes in Alzheimer's disease and mild cognitive impairment," *Neurobiology of aging*, vol. 33, no. 9, pp. 2018–2028, 2012.

[47] S. A. Rombouts, F. Barkhof, R. Goekoop, C. J. Stam, and P. Scheltens, "Altered resting state networks in mild cognitive impairment and mild Alzheimer's disease: an fMRI study," *Human brain mapping*, vol. 26, no. 4, pp. 231–239, 2005.

[48] K. Wang, M. Liang, L. Wang, L. Tian, X. Zhang, K. Li, and T. Jiang, "Altered functional connectivity in early Alzheimer's disease: A resting-state fMRI study," *Human brain mapping*, vol. 28, no. 10, pp. 967–978, 2007.

[49] S. M. Daselaar, S. E. Prince, and R. Cabeza, "When less means more: deactivations during encoding that predict subsequent memory," *Neuroimage*, vol. 23, no. 3, pp. 921–927, 2004.

[50] S. L. Miller, K. Celone, K. DePeau, E. Diamond, B. C. Dickerson, D. Rentz, M. Pihlajamäki, and R. A. Sperling, "Age-related memory impairment associated with loss of parietal deactivation but preserved hippocampal activation," *Proceedings of the National Academy of Sciences*, vol. 105, no. 6, pp. 2181–2186, 2008.

[51] S.-Y. Lin, C.-P. Lin, T.-J. Hsieh, C.-F. Lin, S.-H. Chen, Y.-P. Chao, Y.-S. Chen, C.-C. Hsu, and L.-W. Kuo, "Multiparametric graph theoretical analysis reveals altered structural and functional network topology in Alzheimer's disease," *NeuroImage: Clinical*, vol. 22, p. 101680, 2019.

[52] F. Bai, D. R. Watson, Y. Shi, Y. Wang, C. Yue, YuhuanTeng, D. Wu, Y. Yuan, and Z. Zhang, "Specifically progressive deficits of brain functional marker in amnestic type mild cognitive impairment," *PloS one*, vol. 6, no. 9, p. e24271, 2011.

[53] H. Braak, I. Alafuzoff, T. Arzberger, H. Kretzschmar, and K. Del Tredici, "Staging of Alzheimer disease-associated neurofibrillary pathology using paraffin sections and immunocytochemistry," *Acta neuropathologica*, vol. 112, no. 4, pp. 389–404, 2006.

[54] A. S. Fleisher, A. Sherzai, C. Taylor, J. B. Langbaum, K. Chen, and R. B. Buxton, "Resting-state BOLD networks versus task-associated functional mri for distinguishing Alzheimer's disease risk groups," *Neuroimage*, vol. 47, no. 4, pp. 1678–1690, 2009.

[55] A. Stasenko, D. M. Jacobs, D. P. Salmon, and T. H. Gollan, "The multilingual naming test (MINT) as a measure of picture naming ability in Alzheimer's disease," *Journal of the International Neuropsychological Society*, vol. 25, no. 8, pp. 821–833, 2019.

[56] Y. Chen, Y. Tang, C. Wang, X. Liu, L. Zhao, and Z. Wang, "ADHD classification by dual subspace learning using resting-state functional connectivity," *Artificial intelligence in medicine*, vol. 103, p. 101786, 2020.

[57] Q. Jia, J. Cai, X. Jiang, and S. Li, "A subspace ensemble regression model based slow feature for soft sensing application," *Chinese Journal of Chemical Engineering*, vol. 28, no. 12, pp. 3061–3069, 2020.

[58] W. He, N. Yokoya, and X. Yuan, "Fast hyperspectral image recovery of dual-camera compressive hyperspectral imaging via non-iterative subspace-based fusion," *IEEE Transactions on Image Processing*, vol. 30, pp. 7170–7183, 2021.

[59] Y. Fu, W. Wang, and C. Wang, "Image change detection method based on RPCA and low-rank decomposition," in *2016 35th Chinese Control Conference (CCC)*, pp. 9412–9417, IEEE, 2016.

[60] Z. Zhou, X. Li, J. Wright, E. Candes, and Y. Ma, "Stable principal component pursuit," in *2010 IEEE international symposium on information theory*, pp. 1518–1522, IEEE, 2010.

[61] R. Otazo, E. Candès, and D. K. Sodickson, "Low-rank plus sparse matrix decomposition for accelerated dynamic MRI with separation of background and dynamic components," *Magnetic Resonance in Medicine*, vol. 73, no. 3, pp. 1125–1136, 2015.

[62] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, "Robust subspace learning: Robust PCA, robust subspace tracking, and robust subspace recovery," *IEEE Signal Processing Magazine*, vol. 35, no. 4, pp. 32–55, 2018.

[63] E. P. Agency.

[64] P. Tsalmantza and D. W. Hogg, "A data-driven model for spectra: Finding double redshifts in the sloan digital sky survey," *The Astrophysical Journal*, vol. 753, no. 2, p. 122, 2012.

[65] Y. Cao, A. Zhang, and H. Li, "Multisample estimation of bacterial composition matrices in metagenomics data," *Biometrika*, vol. 107, pp. 75–92, 12 2019.

[66] D. Hong, L. Balzano, and J. A. Fessler, "Asymptotic performance of PCA for high-dimensional heteroscedastic data," *J. Multivar. Anal.*, vol. 167, p. 435–452, sep 2018.

[67] M. E. Tipping and C. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society, Series B*, vol. 21, pp. 611–622, January 1999. Available from http://www.ncrg.aston.ac.uk/Papers/index.html.

[68] D. Hong, K. Gilman, L. Balzano, and J. A. Fessler, "HePPCAT: Probabilistic PCA for data with heteroscedastic noise," *IEEE Transactions on Signal Processing*, vol. 69, pp. 4819–4834, 2021.

[69] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 1–37, 2011.

[70] Y. Chi, Y. M. Lu, and Y. Chen, "Nonconvex optimization meets low-rank matrix factorization: An overview," *IEEE Transactions on Signal Processing*, vol. 67, no. 20, pp. 5239–5269, 2019.

[71] R. Vershynin, *High-dimensional probability: An introduction with applications in data science*, vol. 47. Cambridge university press, 2018.

[72] R. Latała, "Some estimates of norms of random matrices," *Proceedings of the American Mathematical Society*, vol. 133, no. 5, pp. 1273–1282, 2005.

[73] A. R. Zhang, T. T. Cai, and Y. Wu, "Heteroskedastic PCA: Algorithm, optimality, and applications," *The Annals of Statistics*, vol. 50, no. 1, pp. 53–80, 2022.

[74] A. Collas, F. Bouchard, A. Breloy, G. Ginolhac, C. Ren, and J.-P. Ovarlez, "Probabilistic PCA from heteroscedastic signals: geometric framework and application to clustering," *IEEE Transactions on Signal Processing*, vol. 69, pp. 6546–6560, 2021.

[75] L. Delchambre, "Weighted principal component analysis: a weighted covariance eigendecomposition approach," *Monthly Notices of the Royal Astronomical Society*, vol. 446, no. 4, pp. 3545–3555, 2015.

[76] T.-H. Oh, Y.-W. Tai, J.-C. Bazin, H. Kim, and I. S. Kweon, "Partial sum minimization of singular values in robust PCA: Algorithm and applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 4, pp. 744–758, 2015.

[77] N. Johnston and O. Blog, "Ky Fan norms, schatten norms, and everything in between," *Nathaniel Johnston. Np*, vol. 21, 2009.

[78] E. Dobriban, "Permutation methods for factor analysis and PCA," *The Annals of Statistics*, vol. 48, no. 5, pp. 2824 – 2847, 2020.

[79] D. Hong, Y. Sheng, and E. Dobriban, "Selecting the number of components in PCA via random signflips," 2023.

[80] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[81] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *Coordinated Science Laboratory Report no. UILU-ENG-09-2215, DC-247*, 2009.

[82] K. Guo, D. Han, and T.-T. Wu, "Convergence of alternating direction method for minimizing sum of two nonconvex functions with linear constraints," *International Journal of Computer Mathematics*, vol. 94, no. 8, pp. 1653–1669, 2017.

[83] B. Mishra, *Algorithmic algebra*. Springer Science & Business Media, 2012.

[84] H. Attouch, J. Bolte, and B. F. Svaiter, "Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods," *Mathematical Programming*, vol. 137, no. 1-2, pp. 91–129, 2013.

[85] L. van den Dries and C. Miller, "Geometric categories and o-minimal structures," *Duke Mathematical Journal*, vol. 84, no. 2, pp. 497 – 540, 1996.

[86] C. L. Byrne, "Alternating minimization as sequential unconstrained minimization: a survey," *Journal of Optimization Theory and Applications*, vol. 156, pp. 554–566, 2013.

[87] S. Tu, R. Boczar, M. Simchowitz, M. Soltanolkotabi, and B. Recht, "Low-rank solutions of linear matrix equations via Procrustes flow," in *International Conference on Machine Learning*, pp. 964–973, PMLR, 2016.

[88] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[89] O. L. Mangasarian, "Pseudo-convex functions," in *Stochastic optimization models in finance*, pp. 23–32, Elsevier, 1975.

[90] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear Gauss–Seidel method under convex constraints," *Operations research letters*, vol. 26, no. 3, pp. 127–136, 2000.

[91] A. Barg and D. Y. Nogin, "Bounds on packings of spheres in the grassmann manifold," *IEEE Transactions on Information Theory*, vol. 48, no. 9, pp. 2450–2454, 2002.

[92] R. Ahumada, C. A. Prieto, A. Almeida, F. Anders, S. F. Anderson, B. H. Andrews, B. Anguiano, R. Arcodia, E. Armengaud, M. Aubert, *et al.*, "The 16th data release of the sloan digital sky surveys: first release from the APOGEE-2 southern survey and full release of eboss spectra," *The Astrophysical Journal Supplement Series*, vol. 249, no. 1, p. 3, 2020.

[93] B. W. Lyke, A. N. Higley, J. McLane, D. P. Schurhammer, A. D. Myers, A. J. Ross, K. Dawson, S. Chabanier, P. Martini, H. D. M. Des Bourboux, *et al.*, "The sloan digital sky survey quasar catalog: Sixteenth data release," *The Astrophysical Journal Supplement Series*, vol. 250, no. 1, p. 8, 2020.

[94] D. Hong, F. Yang, J. A. Fessler, and L. Balzano, "Optimally weighted PCA for high-dimensional heteroscedastic data," *SIAM Journal on Mathematics of Data Science*, vol. 5, no. 1, pp. 222–250, 2023.

[95] E. Z. Macosko, A. Basu, R. Satija, J. Nemesh, K. Shekhar, M. Goldman, I. Tirosh, A. R. Bialas, N. Kamitaki, E. M. Martersteck, *et al.*, "Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets," *Cell*, vol. 161, no. 5, pp. 1202–1214, 2015.

[96] T. S. Andrews and M. Hemberg, "Identifying cell populations with scRNASeq," *Molecular aspects of medicine*, vol. 59, pp. 114–122, 2018.

[97] P. V. Kharchenko, "The triumphs and limitations of computational methods for scRNA-seq," *Nature methods*, vol. 18, no. 7, pp. 723–732, 2021.

[98] J. Salmon, Z. Harmany, C.-A. Deledalle, and R. Willett, "Poisson noise reduction with non-local PCA," *Journal of mathematical imaging and vision*, vol. 48, pp. 279–294, 2014.

[99] K. Gilman, D. Hong, J. A. Fessler, and L. Balzano, "Streaming probabilistic PCA for missing data with heteroscedastic noise," 2023.

[100] J.-H. Kim, Y. Zhang, K. Han, Z. Wen, M. Choi, and Z. Liu, "Representation learning of resting state fMRI with variational autoencoder," *NeuroImage*, vol. 241, p. 118423, 2021.

[101] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner, "Beta-VAE: Learning basic visual concepts with a constrained variational framework.," *ICLR (Poster)*, vol. 3, 2017.

[102] R. Vidal, "Subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011.

[103] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Unsupervised segmentation of natural images via lossy data compression," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 212–225, 2008.

[104] R. Vidal, R. Tron, and R. Hartley, "Multiframe motion segmentation with missing data using PowerFactorization and GPCA," *International Journal of Computer Vision*, vol. 79, pp. 85–105, 2008.

[105] W. Hong, J. Wright, K. Huang, and Y. Ma, "Multiscale hybrid linear models for lossy image representation," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3655–3671, 2006.

[106] R. Vidal, S. Soatto, Y. Ma, and S. Sastry, "An algebraic geometric approach to the identification of a class of linear hybrid systems," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, vol. 1, pp. 167–172, IEEE, 2003.

[107] R. Heckel and H. Bölcskei, "Robust subspace clustering via thresholding," *IEEE Transactions on Information Theory*, vol. 61, no. 11, pp. 6320–6342, 2015.

[108] J. Lipor, D. Hong, Y. S. Tan, and L. Balzano, "Subspace clustering using ensembles of k-subspaces," *Information and Inference: A Journal of the IMA*, vol. 10, no. 1, pp. 73–107, 2021.

[109] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *International Journal of Computer Vision*, vol. 29, pp. 159–179, 1998.

[110] L. Lu and R. Vidal, "Combined central and subspace clustering for computer vision applications," in *Proceedings of the 23rd international conference on Machine learning*, pp. 593–600, 2006.

[111] S. R. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories," in *2008 IEEE conference on computer vision and pattern recognition*, pp. 1–8, IEEE, 2008.

[112] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 14, 2001.

[113] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, and C. Zhang, "Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering," *IEEE Transactions on Neural Networks*, vol. 22, no. 11, pp. 1796–1808, 2011.

[114] W. Qu, X. Xiu, H. Chen, and L. Kong, "A survey on high-dimensional subspace clustering," *Mathematics*, vol. 11, no. 2, p. 436, 2023.

[115] R. Vidal and P. Favaro, "Low rank subspace clustering (LRSC)," *Pattern Recognition Letters*, vol. 43, pp. 47–61, 2014.

[116] P. S. Bradley and O. L. Mangasarian, "K-plane clustering," *Journal of Global optimization*, vol. 16, pp. 23–32, 2000.

[117] P. Wang, H. Liu, A. M.-C. So, and L. Balzano, "Convergence and recovery guarantees of the k-subspaces method for subspace clustering," in *International Conference on Machine Learning*, pp. 22884–22918, PMLR, 2022.

[118] A. Gitlin, B. Tao, L. Balzano, and J. Lipor, "Improving k-subspaces via coherence pursuit," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 6, pp. 1575–1588, 2018.

[119] M. E. Timmerman, E. Ceulemans, K. De Roover, and K. Van Leeuwen, "Subspace k-means clustering," *Behavior research methods*, vol. 45, pp. 1011–1023, 2013.

[120] J. Ghosh and A. Acharya, "Cluster ensembles," *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, vol. 1, no. 4, pp. 305–315, 2011.

[121] R. A. Lobos, J. Salazar Cavazos, R. R. Nadakuditi, and J. A. Fessler, "Smooth optimization algorithms for global and locally low-rank regularizers," *arXiv preprint arXiv:2505.06073*, 2025.

[122] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the l2-graph for robust subspace learning and subspace clustering," *IEEE transactions on cybernetics*, vol. 47, no. 4, pp. 1053–1066, 2016.

[123] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[124] D. Lim, R. Vidal, and B. D. Haeffele, "Doubly stochastic subspace clustering," *arXiv preprint arXiv:2011.14859*, 2020.

[125] M. F. Baumgardner, L. L. Biehl, and D. A. Landgrebe, "220 band aviris hyperspectral image data set: June 12, 1992 indian pine test site 3," Sep 2015.

[126] A. Mahmood and M. Sears, "Per-pixel noise estimation in hyperspectral images," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

[127] S. Zhang, X. Kang, Y. Mo, and S. Li, "Noise analysis of hyperspectral images captured by different sensors," in *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*, pp. 2707–2710, IEEE, 2020.

[128] D. Uchaev and D. Uchaev, "Small sample hyperspectral image classification based on the random patches network and recursive filtering," *Sensors*, vol. 23, no. 5, p. 2499, 2023.

[129] S. Huang, H. Zhang, Q. Du, and A. Pi$v$zurica, "Sketch-based subspace clustering of hyperspectral images," *Remote Sensing*, vol. 12, no. 5, p. 775, 2020.

[130] R. Sperling, "The potential of functional MRI as a biomarker in early Alzheimer's disease," *Neurobiology of aging*, vol. 32, pp. S37–S43, 2011.

[131] R. K. Gupta and J. Kuznicki, "Biological and medical importance of cellular heterogeneity deciphered by single-cell rna sequencing," *Cells*, vol. 9, no. 8, p. 1751, 2020.

[132] R. M. Larsen, "Lanczos bidiagonalization with partial reorthogonalization," *DAIMI Report Series*, vol. 27, no. 537, 1998.

[133] C. You, C. Li, D. P. Robinson, and R. Vidal, "Scalable exemplar-based subspace clustering on class-imbalanced data," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[134] J. Qian and V. Saligrama, "Spectral clustering with imbalanced data," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3057–3061, IEEE, 2014.

[135] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[136] T. Darcet, M. Oquab, J. Mairal, and P. Bojanowski, "Vision transformers need registers," in *The Twelfth International Conference on Learning Representations*, ICLR, 2024.

[137] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, "DINOv2: Learning robust visual features without supervision," *Transactions on Machine Learning Research Journal*, pp. 1–31, 2024.

[138] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, ICLR, 2022.

[139] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[140] H. Hu, X. Wang, Y. Zhang, Q. Chen, and Q. Guan, "A comprehensive survey on contrastive learning," *Neurocomputing*, p. 128645, 2024.

[141] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, and T. Kong, "iBot: Image BERT pre-training with online tokenizer," in *International Conference on Learning Representations*, ICLR, 2022.

[142] A. Gadetsky, Y. Jiang, and M. Brbic, "Let go of your labels with unsupervised transfer," in *International Conference on Machine Learning*, pp. 14382–14407, PMLR, 2024.

[143] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International conference on machine learning*, pp. 478–487, ICML, 2016.

[144] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep adaptive image clustering," in *Proceedings of the IEEE international conference on computer vision*, pp. 5879–5887, 2017.

[145] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 132–149, 2018.

[146] C. Niu, H. Shan, and G. Wang, "Spice: Semantic pseudo-labeling for image clustering," *IEEE Transactions on Image Processing*, vol. 31, pp. 7264–7278, 2022.

[147] A. Mao, M. Mohri, and Y. Zhong, "Cross-entropy loss functions: Theoretical analysis and applications," in *International conference on Machine learning*, pp. 23803–23828, pmlr, 2023.

[148] D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro, "The implicit bias of gradient descent on separable data," *Journal of Machine Learning Research*, vol. 19, no. 70, pp. 1–57, 2018.

[149] Z. Ji and M. Telgarsky, "The implicit bias of gradient descent on nonseparable data," in *Conference on learning theory*, pp. 1772–1798, PMLR, 2019.

[150] A. Gadetsky and M. Brbic, "The pursuit of human labeling: a new perspective on unsupervised learning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[151] M. Assran, R. Balestriero, Q. Duval, F. Bordes, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, and N. Ballas, "The hidden uniform cluster prior in self-supervised learning," in *International Conference on Learning Representations*, ICLR, 2023.

[152] H. Zhou, T. Luo, and Y. He, "Dynamic collaborative learning with heterogeneous knowledge transfer for long-tailed visual recognition," *Information Fusion*, vol. 115, p. 102734, 2025.

[153] X. Zhang and T. Luo, "Imbalanced multi-instance multi-label learning via tensor product-based semantic fusion," *Frontiers of Computer Science*, vol. 19, no. 8, p. 198346, 2025.

[154] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[155] H. Zhou, T. Luo, J. Zhang, and L. Liu, "Exploring the essence of relationships for scene graph generation via causal features enhancement network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[156] A. Clauset, C. R. Shalizi, and M. E. Newman, "Power-law distributions in empirical data," *SIAM review*, vol. 51, no. 4, pp. 661–703, 2009.

[157] M. Mahajan, P. Nimbhorkar, and K. Varadarajan, "The planar k-means problem is np-hard," *Theoretical computer science*, vol. 442, pp. 13–21, 2012.

[158] A. Martins and R. Astudillo, "From softmax to sparsemax: A sparse model of attention and multi-label classification," in *International conference on machine learning*, pp. 1614–1623, PMLR, 2016.

[159] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l 1-ball for learning in high dimensions," in *Proceedings of the 25th international conference on Machine learning*, pp. 272–279, 2008.

[160] D. Arthur and S. Vassilvitskii, "K-Means++ the advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035, 2007.

[161] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, pp. 1597–1607, PmLR, 2020.

[162] A. Krizhevsky, "Learning multiple layers of features from tiny images," *Master's thesis, University of Toronto*, 2009.

[163] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101–mining discriminative components with random forests," in *European conference on computer vision*, pp. 446–461, Springer, 2014.

[164] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 4, pp. 594–611, 2006.

[165] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

[166] P. Helber, B. Bischke, A. Dengel, and D. Borth, "EuroSat: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 7, pp. 2217–2226, 2019.

[167] A. Acevedo, A. Merino, S. Alférez, Á. Molina, L. Boldú, and J. Rodellar, "A dataset of microscopic peripheral blood cell images for development of automatic recognition systems," *Data in brief*, vol. 30, p. 105474, 2020.

[168] P. Tschandl, C. Rosendahl, and H. Kittler, "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Scientific data*, vol. 5, no. 1, pp. 1–9, 2018.

[169] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie, "The inaturalist species classification and detection dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8769–8778, 2018.

[170] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan, *et al.*, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *cell*, vol. 172, no. 5, pp. 1122–1131, 2018.

[171] P. Bilic, P. Christ, H. B. Li, E. Vorontsov, A. Ben-Cohen, G. Kaissis, A. Szeskin, C. Jacobs, G. E. H. Mamani, G. Chartrand, *et al.*, "The liver tumor segmentation benchmark (lits)," *Medical image analysis*, vol. 84, p. 102680, 2023.

[172] V. Ljosa, K. L. Sokolnicki, and A. E. Carpenter, "Annotated high-throughput microscopy image sets for validation," *Nature methods*, vol. 9, no. 7, p. 637, 2012.

[173] J. Yang, R. Shi, D. Wei, Z. Liu, L. Zhao, B. Ke, H. Pfister, and B. Ni, "MedMNIST v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification," *Scientific Data*, vol. 10, no. 1, p. 41, 2023.

[174] W. F. Wiggins and A. S. Tejani, "On the opportunities and risks of foundation models for natural language processing in radiology," *Radiology: Artificial Intelligence*, vol. 4, no. 4, p. e220119, 2022.

[175] P. Vemuri, D. T. Jones, and C. R. Jack, "Resting state functional MRI in Alzheimer's disease," *Alzheimer's research & therapy*, vol. 4, pp. 1–9, 2012.

[176] M. Karker, *Predictive Analysis and Deep Learning of Functional MRI in Alzheimer's Disease*. PhD thesis, University of Michigan, 2022.

[177] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022.

[178] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[179] A. Myronenko, D. Yang, V. Buch, D. Xu, A. Ihsani, S. Doyle, M. Michalski, N. Tenenholtz, and H. Roth, "4D CNN for semantic segmentation of cardiac volumetric sequences," in *Statistical Atlases and Computational Models of the Heart. Multi-Sequence CMR Segmentation, CRT-EPiggy and LV Full Quantification Challenges: 10th International Workshop, STACOM 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Revised Selected Papers 10*, pp. 72–80, Springer, 2020.

[180] M. Bengs, N. Gessert, and A. Schlaefer, "4d spatio-temporal deep learning with 4d fmri data for autism spectrum disorder classification," *arXiv preprint arXiv:2004.10165*, 2020.

[181] Y. Dogan, "A new global pooling method for deep neural networks: Global average of top-k max-pooling," *Traitement du signal*, vol. 40, no. 2, pp. 577–587, 2023.

[182] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[183] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[184] K. J. Gorgolewski, T. Auer, V. D. Calhoun, R. C. Craddock, S. Das, E. P. Duff, G. Flandin, S. S. Ghosh, T. Glatard, Y. O. Halchenko, *et al.*, "The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[185] O. Esteban, C. J. Markiewicz, R. W. Blair, C. A. Moodie, A. I. Isik, A. Erramuzpe, J. D. Kent, M. Goncalves, E. DuPre, M. Snyder, *et al.*, "fMRIPrep: a robust preprocessing pipeline for functional MRI," *Nature methods*, vol. 16, no. 1, pp. 111–116, 2019.

[186] R. Ciric, W. H. Thompson, R. Lorenz, M. Goncalves, E. E. MacNicol, C. J. Markiewicz, Y. O. Halchenko, S. S. Ghosh, K. J. Gorgolewski, R. A. Poldrack, *et al.*, "TemplateFlow: FAIR-sharing of multi-scale, multi-species brain models," *Nature Methods*, vol. 19, no. 12, pp. 1568–1571, 2022.

[187] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[188] M. Brett, C. J. Markiewicz, M. Hanke, M.-A. Côté, B. Cipollini, P. McCarthy, D. Jarecka, C. P. Cheng, Y. O. Halchenko, M. Cottaar, *et al.*, "nipy/nibabel: 3.2. 1," *Zenodo*, 2020.

[189] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[190] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-Cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

[191] D. A. Orringer, D. R. Vago, and A. J. Golby, "Clinical applications and future directions of functional MRI," in *Seminars in neurology*, vol. 32, pp. 466–475, Thieme Medical Publishers, 2012.

[192] R. L. Buckner, F. M. Krienen, and B. T. Yeo, "Opportunities and limitations of intrinsic functional connectivity MRI," *Nature neuroscience*, vol. 16, no. 7, pp. 832–837, 2013.

[193] S. J. Peltier and Y. Shah, "Biophysical modulations of functional connectivity," *Brain connectivity*, vol. 1, no. 4, pp. 267–277, 2011.

[194] E. M. Lake, E. S. Finn, S. M. Noble, T. Vanderwal, X. Shen, M. D. Rosenberg, M. N. Spann, M. M. Chun, D. Scheinost, and R. T. Constable, "The functional brain organization of an individual allows prediction of measures of social abilities transdiagnostically in autism and attention-deficit/hyperactivity disorder," *Biological psychiatry*, vol. 86, no. 4, pp. 315–326, 2019.

[195] M. D. Rosenberg, E. S. Finn, D. Scheinost, X. Papademetris, X. Shen, R. T. Constable, and M. M. Chun, "A neuromarker of sustained attention from whole-brain functional connectivity," *Nature neuroscience*, vol. 19, no. 1, pp. 165–171, 2016.

[196] M. C. Kroes, M. D. Rugg, M. G. Whalley, and C. R. Brewin, "Structural brain abnormalities common to posttraumatic stress disorder and depression," *Journal of Psychiatry and Neuroscience*, vol. 36, no. 4, pp. 256–265, 2011.

[197] K. Zukotynski, V. Gaudet, P. H. Kuo, S. Adamo, M. Goubran, C. J. Scott, C. Bocti, M. Borrie, H. Chertkow, R. Frayne, *et al.*, "The use of random forests to identify brain regions on amyloid and FDG-PET associated with MoCA score," *Clinical nuclear medicine*, vol. 45, no. 6, pp. 427–433, 2020.

[198] C. Liu, L. Li, D. Zhu, S. Lin, L. Ren, W. Zhen, W. Tan, L. Wang, L. Tian, Q. Wang, *et al.*, "Individualized prediction of cognitive test scores from functional brain connectome in patients with first-episode late-life depression," *Journal of Affective Disorders*, vol. 352, pp. 32–42, 2024.

[199] X. Shen, E. S. Finn, D. Scheinost, M. D. Rosenberg, M. M. Chun, X. Papademetris, and R. T. Constable, "Using connectome-based predictive modeling to predict individual behavior from brain connectivity," *nature protocols*, vol. 12, no. 3, pp. 506–518, 2017.

[200] Q. Lin, M. D. Rosenberg, K. Yoo, T. W. Hsu, T. P. O'Connell, and M. M. Chun, "Resting-state functional connectivity predicts cognitive impairment related to Alzheimer's disease," *Frontiers in aging neuroscience*, vol. 10, p. 94, 2018.

[201] W. G. Rosen, R. C. Mohs, and K. L. Davis, "A new rating scale for Alzheimer's disease.," *The American journal of psychiatry*, vol. 141, no. 11, pp. 1356–1364, 1984.

[202] B. M. Hampstead, A. Iordan, A. Rahman-Filipiak, R. Ploutz-Snyder, and P. Pruitt, "Synergistic effects of cognitive training and high-definition transcranial direct current stimulation across the dementia spectrum," *Brain Stimulation: Basic, Translational, and Clinical Research in Neuromodulation*, vol. 18, no. 1, p. 350, 2025.

[203] S. Weintraub, D. Salmon, N. Mercaldo, S. Ferris, N. R. Graff-Radford, H. Chui, J. Cummings, C. DeCarli, N. L. Foster, D. Galasko, *et al.*, "The Alzheimer's disease centers' uniform data set (UDS): the neuropsychologic test battery," *Alzheimer Disease & Associated Disorders*, vol. 23, no. 2, pp. 91–101, 2009.

[204] C. O. Nester, J. Qin, C. Wang, M. J. Katz, R. B. Lipton, and L. A. Rabin, "Concordance between logical memory and craft story 21 in community-dwelling older adults: the role of demographic factors and cognitive status," *Archives of Clinical Neuropsychology*, vol. 38, no. 7, pp. 1091–1105, 2023.

[205] S. Whitfield-Gabrieli and A. Nieto-Castanon, "CONN: a functional connectivity toolbox for correlated and anticorrelated brain networks," *Brain connectivity*, vol. 2, no. 3, pp. 125–141, 2012.

[206] J. D. Power, A. L. Cohen, S. M. Nelson, G. S. Wig, K. A. Barnes, J. A. Church, A. C. Vogel, T. O. Laumann, F. M. Miezin, B. L. Schlaggar, *et al.*, "Functional network organization of the human brain," *Neuron*, vol. 72, no. 4, pp. 665–678, 2011.

[207] G. C. McDonald, "Ridge regression," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 1, no. 1, pp. 93–100, 2009.

[208] V. Vovk, "Kernel ridge regression," in *Empirical inference: Festschrift in honor of vladimir n. vapnik*, pp. 105–116, Springer, 2013.

[209] B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[210] E. S. Finn, X. Shen, D. Scheinost, M. D. Rosenberg, J. Huang, M. M. Chun, X. Papademetris, and R. T. Constable, "Functional connectome fingerprinting: identifying individuals using patterns of brain connectivity," *Nature neuroscience*, vol. 18, no. 11, pp. 1664–1671, 2015.

[211] A. S. Greene, S. Gao, D. Scheinost, and R. T. Constable, "Task-induced brain state manipulation improves prediction of individual traits," *Nature communications*, vol. 9, no. 1, p. 2807, 2018.

[212] D. Langlois, S. Chartier, and D. Gosselin, "An introduction to independent component analysis: InfoMax and FastICA algorithms," *Tutorials in Quantitative Methods for Psychology*, vol. 6, no. 1, pp. 31–38, 2010.

[213] V. D. Calhoun, T. Adali, G. D. Pearlson, and J. J. Pekar, "A method for making group inferences from functional MRI data using independent component analysis," *Human brain mapping*, vol. 14, no. 3, pp. 140–151, 2001.

[214] C. F. Beckmann, M. DeLuca, J. T. Devlin, and S. M. Smith, "Investigations into resting-state connectivity using independent component analysis," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 360, no. 1457, pp. 1001–1013, 2005.

[215] L. Griffanti, G. Douaud, J. Bijsterbosch, S. Evangelisti, F. Alfaro-Almagro, M. F. Glasser, E. P. Duff, S. Fitzgibbon, R. Westphal, D. Carone, *et al.*, "Hand classification of fMRI ICA noise components," *Neuroimage*, vol. 154, pp. 188–205, 2017.

[216] V. D. Calhoun, J. Liu, and T. Adalı, "A review of group ICA for fMRI data and ICA for joint inference of imaging, genetic, and ERP data," *Neuroimage*, vol. 45, no. 1, pp. S163–S172, 2009.

[217] V. J. Schmithorst and S. K. Holland, "Comparison of three methods for generating group statistical inferences from independent component analysis of functional magnetic resonance imaging data," *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 19, no. 3, pp. 365–368, 2004.

[218] L. Yang, Y. Yan, Y. Wang, X. Hu, J. Lu, P. Chan, T. Yan, and Y. Han, "Gradual disturbances of the amplitude of low-frequency fluctuations (ALFF) and fractional ALFF in Alzheimer spectrum," *Frontiers in neuroscience*, vol. 12, p. 975, 2018.

[219] J.-J. Wang, X. Chen, S. Sah, C. Zeng, Y.-M. Li, N. Li, M.-Q. Liu, and S.-L. Du, "Amplitude of low-frequency fluctuation (ALFF) and fractional ALFF in migraine patients: a resting-state functional MRI study," *Clinical radiology*, vol. 71, no. 6, pp. 558–564, 2016.

[220] R. Yu, Y.-L. Chien, H.-L. S. Wang, C.-M. Liu, C.-C. Liu, T.-J. Hwang, M. H. Hsieh, H.-G. Hwu, and W.-Y. I. Tseng, "Frequency-specific alternations in the amplitude of low-frequency fluctuations in schizophrenia," *Human brain mapping*, vol. 35, no. 2, pp. 627–637, 2014.

[221] Q.-H. Zou, C.-Z. Zhu, Y. Yang, X.-N. Zuo, X.-Y. Long, Q.-J. Cao, Y.-F. Wang, and Y.-F. Zang, "An improved approach to detection of amplitude of low-frequency fluctuation (ALFF) for resting-state fMRI: fractional ALFF," *Journal of neuroscience methods*, vol. 172, no. 1, pp. 137–141, 2008.

[222] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," in *The Sixth International Conference on Learning Representations (ICLR)*, 2018.

[223] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[224] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional LSTM networks for improved phoneme classification and recognition," in *International conference on artificial neural networks*, pp. 799–804, Springer, 2005.

[225] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

[226] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[227] M. Aoki, *State space modeling of time series*. Springer Science & Business Media, 2013.

[228] A. Gu, K. Goel, and C. Re, "Efficiently modeling long sequences with structured state spaces," in *International Conference on Learning Representations*, 2022.

[229] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," in *International Conference on Learning Representations*, 2024.

[230] T. Dao and A. Gu, "Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality," in *International Conference on Machine Learning*, pp. 10041–10071, PMLR, 2024.

[231] G. E. Blelloch, "Prefix sums and their applications," *Carnegie Mellon University Pittsburgh, PA, USA*, 1990.

[232] A. Gu, T. Dao, S. Ermon, A. Rudra, and C. Ré, "HIPPO: Recurrent memory with optimal polynomial projections," *Advances in neural information processing systems*, vol. 33, pp. 1474–1487, 2020.

[233] T. Ye, L. Dong, Y. Xia, Y. Sun, Y. Zhu, G. Huang, and F. Wei, "Differential transformer," in *The Thirteenth International Conference on Learning Representations*, 2025.

[234] N. Schneider, I. Zimerman, and E. Nachmani, "Differential mamba," in *Association for Computational Linguistics (AACL)*, 2025.

[235] D. Lei, Z. Sun, Y. Xiao, and W. Y. Wang, "Implicit regularization of stochastic gradient descent in natural language processing: Observations and implications," *arXiv preprint arXiv:1811.00659*, 2018.

[236] D. Masters and C. Luschi, "Revisiting small batch training for deep neural networks," *arXiv preprint arXiv:1804.07612*, 2018.

[237] A. Sekhari, K. Sridharan, and S. Kale, "SGD: The role of implicit regularization, batch-size and multiple-epochs," *Advances In Neural Information Processing Systems*, vol. 34, pp. 27422–27433, 2021.

[238] B. Zhang and R. Sennrich, "Root mean square layer normalization," *Advances in neural information processing systems*, vol. 32, 2019.

[239] A. Altmann, L. Tolosi, O. Sander, and T. Lengauer, "Permutation importance: a corrected feature importance measure," *Bioinformatics*, vol. 26, no. 10, pp. 1340–1347, 2010.

[240] J. L. Lancaster, M. G. Woldorff, L. M. Parsons, M. Liotti, C. S. Freitas, L. Rainey, P. V. Kochunov, D. Nickerson, S. A. Mikiten, and P. T. Fox, "Automated talairach atlas labels for functional brain mapping," *Human brain mapping*, vol. 10, no. 3, pp. 120–131, 2000.

[241] A. M. Ward, A. P. Schultz, W. Huijbers, K. R. Van Dijk, T. Hedden, and R. A. Sperling, "The parahippocampal gyrus links the default-mode cortical network with the medial temporal lobe memory system," *Human brain mapping*, vol. 35, no. 3, pp. 1061–1073, 2014.

[242] A. Du, N. Schuff, D. Amend, M. Laakso, Y. Hsu, W. Jagust, K. Yaffe, J. Kramer, B. Reed, D. Norman, *et al.*, "Magnetic resonance imaging of the entorhinal cortex and hippocampus in mild cognitive impairment and Alzheimer's disease," *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 71, no. 4, pp. 441–447, 2001.

[243] K. A. Celone, V. D. Calhoun, B. C. Dickerson, A. Atri, E. F. Chua, S. L. Miller, K. DePeau, D. M. Rentz, D. J. Selkoe, D. Blacker, *et al.*, "Alterations in memory networks in mild cognitive impairment and Alzheimer's disease: an independent component analysis," *Journal of Neuroscience*, vol. 26, no. 40, pp. 10222–10231, 2006.

[244] H. Braak and E. Braak, "Neuropathological stageing of Alzheimer-related changes," *Acta neuropathologica*, vol. 82, no. 4, pp. 239–259, 1991.

[245] H. Yang, H. Xu, Q. Li, Y. Jin, W. Jiang, J. Wang, Y. Wu, W. Li, C. Yang, X. Li, *et al.*, "Study of brain morphology change in Alzheimer's disease and amnestic mild cognitive impairment compared with normal controls," *General psychiatry*, vol. 32, no. 2, p. e100005, 2019.

[246] M. De Marco, D. Duzzi, F. Meneghello, and A. Venneri, "Cognitive efficiency in Alzheimer's disease is associated with increased occipital connectivity," *Journal of Alzheimer's Disease*, vol. 57, no. 2, pp. 541–556, 2017.

[247] A. E. Cavanna and M. R. Trimble, "The precuneus: a review of its functional anatomy and behavioural correlates," *Brain*, vol. 129, no. 3, pp. 564–583, 2006.

[248] M. Bailly, C. Destrieux, C. Hommet, K. Mondon, J.-P. Cottier, E. Beaufils, E. Vierron, J. Vercouillie, M. Ibazizene, T. Voisin, *et al.*, "Precuneus and cingulate cortex atrophy and hypometabolism in patients with Alzheimer's disease and mild cognitive impairment: MRI and 18F-FDG PET quantitative analysis using FreeSurfer," *BioMed research international*, vol. 2015, no. 1, p. 583931, 2015.

[249] G. Karas, P. Scheltens, S. Rombouts, R. Van Schijndel, M. Klein, B. Jones, W. Van Der Flier, H. Vrenken, and F. Barkhof, "Precuneus atrophy in early-onset Alzheimer's disease: a morphometric structural MRI study," *Neuroradiology*, vol. 49, no. 12, pp. 967–976, 2007.

[250] F. C. Binkofski, J. Klann, and S. Caspers, "On the neuroanatomy and functional role of the inferior parietal lobule and intraparietal sulcus," in *Neurobiology of language*, pp. 35–47, Elsevier, 2016.

[251] S. J. Greene, R. J. Killiany, Alzheimer's Disease Neuroimaging Initiative, *et al.*, "Subregions of the inferior parietal lobule are affected in the progression to Alzheimer's disease," *Neurobiology of aging*, vol. 31, no. 8, pp. 1304–1311, 2010.

[252] Z. Wang, M. Xia, Z. Dai, X. Liang, H. Song, Y. He, and K. Li, "Differentially disrupted functional connectivity of the subregions of the inferior parietal lobule in Alzheimer's disease," *Brain Structure and Function*, vol. 220, no. 2, pp. 745–762, 2015.

[253] F. L. Stevens, R. A. Hurley, and K. H. Taber, "Anterior cingulate cortex: unique role in cognition and emotion," *The Journal of neuropsychiatry and clinical neurosciences*, vol. 23, no. 2, pp. 121–125, 2011.

[254] H.-J. Jeong, Y.-M. Lee, J.-M. Park, B.-D. Lee, E. Moon, H. Suh, H.-J. Kim, K. Pak, K.-U. Choi, and Y.-I. Chung, "Reduced thickness of the anterior cingulate cortex as a predictor of amnestic-mild cognitive impairment conversion to Alzheimer's disease with psychosis," *Journal of Alzheimer's Disease*, vol. 84, no. 4, pp. 1709–1717, 2021.

[255] X. Liu, W. Chen, H. Hou, X. Chen, J. Zhang, J. Liu, Z. Guo, and G. Bai, "Decreased functional connectivity between the dorsal anterior cingulate cortex and lingual gyrus in Alzheimer's disease patients with depression," *Behavioural brain research*, vol. 326, pp. 132–138, 2017.

[256] S. Tekin, M. S. Mega, D. M. Masterman, T. Chow, J. Garakian, H. V. Vinters, and J. L. Cummings, "Orbitofrontal and anterior cingulate cortex neurofibrillary tangle burden is associated with agitation in Alzheimer disease," *Annals of neurology*, vol. 49, no. 3, pp. 355–361, 2001.

[257] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[258] F. de Vos, M. Koini, T. M. Schouten, S. Seiler, J. van der Grond, A. Lechner, R. Schmidt, M. de Rooij, and S. A. Rombouts, "A comprehensive analysis of resting state fMRI measures to classify individual patients with Alzheimer's disease," *Neuroimage*, vol. 167, pp. 62–72, 2018.

[259] A. Alorf and M. U. G. Khan, "Multi-label classification of Alzheimer's disease stages from resting-state fMRI-based correlation connectivity data and deep learning," *Computers in Biology and Medicine*, vol. 151, p. 106240, 2022.

[260] F. Ramzan, M. U. G. Khan, A. Rehmat, S. Iqbal, T. Saba, A. Rehman, and Z. Mehmood, "A deep learning approach for automated diagnosis and multi-class classification of Alzheimer's disease stages using resting-state fMRI and residual neural networks," *Journal of medical systems*, vol. 44, no. 2, p. 37, 2020.

[261] U. Khatri and G.-R. Kwon, "Alzheimer's disease diagnosis and biomarker analysis using resting-state functional MRI functional brain network with multi-measures features and hippocampal subfield and amygdala volume of structural MRI," *Frontiers in aging neuroscience*, vol. 14, p. 818871, 2022.

[262] J.-H. Noh, J.-H. Kim, and H.-D. Yang, "Classification of Alzheimer's progression using fMRI data," *Sensors*, vol. 23, no. 14, p. 6330, 2023.

[263] Q.-L. He, Y. Zhou, Y. Liu, X.-Q. Li, S.-K. Zhao, Q. Xie, G. Feng, and J.-X. Wang, "Parameter-determined effects: Advances in transcranial focused ultrasound for modulating neural excitation and inhibition," *Bioengineering*, vol. 13, no. 1, p. 20, 2025.

[264] B. Sun, J. Feng, and K. Saenko, "Correlation alignment for unsupervised domain adaptation," in *Domain adaptation in computer vision applications*, pp. 153–171, Springer, 2017.

[265] D. M. Rentz, R. E. Amariglio, J. A. Becker, M. Frey, L. E. Olson, K. Frishe, J. Carmasin, J. E. Maye, K. A. Johnson, and R. A. Sperling, "Face-name associative memory performance is related to amyloid burden in normal elderly," *Neuropsychologia*, vol. 49, no. 9, pp. 2776–2783, 2011.

[266] B. M. Hampstead, A. Y. Stringer, R. F. Stilla, A. Amaraneni, and K. Sathian, "Where did i put that? patients with amnestic mild cognitive impairment demonstrate widespread reductions in activity during the encoding of ecologically relevant object-location associations," *Neuropsychologia*, vol. 49, no. 9, pp. 2349–2361, 2011.

[267] B. M. Hampstead, K. Sathian, M. Bikson, and A. Y. Stringer, "Combined mnemonic strategy training and high-definition transcranial direct current stimulation for memory deficits in mild cognitive impairment," *Alzheimer's & Dementia: Translational Research & Clinical Interventions*, vol. 3, no. 3, pp. 459–470, 2017.

[268] B. M. Hampstead, A. Y. Stringer, R. F. Stilla, G. Deshpande, X. Hu, A. B. Moore, and K. Sathian, "Activation and effective connectivity changes following explicit-memory training for face–name pairs in patients with mild cognitive impairment: a pilot study," *Neurorehabilitation and neural repair*, vol. 25, no. 3, pp. 210–222, 2011.

[269] A. Krishnan, L. J. Williams, A. R. McIntosh, and H. Abdi, "Partial least squares (PLS) methods for neuroimaging: a tutorial and review," *Neuroimage*, vol. 56, no. 2, pp. 455–475, 2011.

[270] B. Landa and Y. Kluger, "The Dyson equalizer: Adaptive noise stabilization for low-rank signal detection and recovery," *Information and Inference: A Journal of the IMA*, vol. 14, no. 1, p. iaae036, 2025.

[271] A. L. MacKay and C. Laule, "Magnetic resonance of myelin water: an in vivo marker for myelin," *Brain plasticity*, vol. 2, no. 1, pp. 71–91, 2016.