中图分类号：TP391 论文编号：1028716 18-B008
学科分类号：081200

博士学位论文

脑影像功能校准与特征学习研究及应用

研究生姓名	Muhammad Yousefnezhad
学科、专业	计算机科学与技术
研究方向	模式识别与应用
指导教师	张道强　　教授

南京航空航天大学
研究生院　计算机科学与技术学院
二Ｏ一八年六月

Nanjing University of Aeronautics and Astronautics
The Graduate School
College of Computer Science and Technology

Functional Alignment and Feature Learning with Neuroimaging Data

A Thesis in

College of Computer Science and Technology

by

Muhammad Yousefnezhad

Advised by

Prof. Daoqiang Zhang

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

June, 2018

To my family.

承诺书

本人声明所呈交的博士学位论文是本人在导师指导下进行的研究工作及取得的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得南京航空航天大学或其他教育机构的学位或证书而使用过的材料。

本人授权南京航空航天大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后适用本承诺书）

作者签名：__________

日　　期：__________

摘　要

目前在神经科学和机器学习领域中的一个重大的挑战是如何正确地理解人类大脑的工作方式，其中对大脑中思想、记忆和情感等功能的研究将促进包括科学、医学、教育等领域的发展。功能性神经影像作为一种成像技术能够用来表示测量到的神经活动，其主要思想是利用这些影像数据来揭示认知过程。事实上，借助这些影像数据，我们不仅能够了解人类大脑区域的功能，而且能够研究这些区域所代表的具体信息以及该信息是如何编码的。神经活动可以从不同层面进行分析，其中最关键的步骤是了解不同的认知任务之间的相似性（或差异性），进而建立认知模型来分析神经活动，这能够帮助我们增加对人脑的认知并促进精神疾病治疗的发展。

过去十五年来，人们在研究人类神经活动解码方面取得了一些重大的进展，然而，仍然有几个长期存在的挑战，其中包括多被试者数据的功能校准、有判别性的特征的选择、无监督相似性分析以及用于预测神经活动的监督模型的生成等问题。本文提出新的方法来分析不同层次和应用中的认知过程，主要贡献可概括如下:

(1) 本文提出了两种功能校准技术，即无监督和监督方法。首先，本文提出深度超校准（Deep Hyperalignment, DHA）方法作为功能校准的新型无监督方法。DHA 采用深度核函数，包括一个多层神经网络，它可以实现任意非线性函数，而且它使用秩为 m 的奇异值分解（rank-m Singular Value Decomposition, SVD）和随机梯度下降（Stochastic Gradient Descent, SGD）进行优化。此外，当把训练好的 DHA 模型运用到新的被试者数据上时，不需要使用训练数据。因此，DHA 在大型数据集上的运行时间较少。作为一个有监督的可行方法，局部判别超校准（Local Discriminant Hyperalignment, LDHA）方法通过将局部判别分析（Local Discriminate Analysis, LDA）的思想融入到典型相关分析（Canonical Correlation Analysis, CCA）中来提高超校准的性能。事实上，局部性的概念是基于训练集中的刺激类别（类别标签）来定义的，该方法首先为每个类别的刺激生成两个集合，即作为类内邻域的最近同类刺激集合和作为类间邻域的不同类别刺激集合，然后我们最大化类内邻域之间的相关性并且使得类间邻域之间的相关性接近于零，以此来产生更好的超校准的解。最后，我们发现在二分类问题上 DHA 具有较好的性能，然而在多分类问题上 LDHA 的性能更好。

(2) 本文提出了新的特征分析技术。所提出的方法为每个刺激选择大脑图像的一个快照而不是分析整个时间序列。经典方法仅仅从体素空间中提取特征，而本文所提出的方法在解码神经活动时可以选择时间序列的一个子集，实际上，我们可以通过在平滑后的设计矩阵中查找局部最大值来选择这些快照。最后，本文提出两种学习方法。事实上，可以使用无监督学习和监督学习两种方法来分析提取的特征。为了应用无监督学习，本论文提出了一种集群集成方法，在该方法中，神经活动之间的相似性或距离可以跨被试者进行比较。对于监督学习方法，本文提出了基于 $\ell 1$ 正则化的支持向量机（Support Vector Machine , SVM）二分类器的 Bagging 技术，在该技术中每个分类器由每个大脑解剖区域中的神经活动生成。这种方法的主要贡献在于其可以减少功能性磁共振成像（functional Magnetic Resonance Imaging, fMRI）数据集的稀疏性。

(3) 本文提出了深度表征相似性分析（Deep Representational Similarity Analysis, DRSA）方法来进行相似性分析。DRSA 使用深度核函数把非线性神经活动转换到线性嵌入空间中去，然后在该空间中评估映射后的特征之间的相似性（或距离）。此外，为了在不同类别的刺激的相关性和协方差之间取得平衡，该方法使用了一个新的正则化项。由于 DRSA 采用基于梯度的优化方法，因此评估高维神经影像数据（如全脑图像）的时间效率较高。

(4) 此外，本文还提出了一种新的监督学习技术。首先，本文提出不平衡 AdaBoost 二分类（Imbalance AdaBoost Binary Classification, IABC）方法作为不平衡二分类学习的一种新技术，该种方法适合于一对多分类分析。事实上为了提高预测性能，IABC 使用监督随机抽样和惩罚值，其中惩罚值通过不同类别之间的相关性计算得到。随后通过使用纠错输出编（Error-Correcting Output Codes, ECOC）方法对 IABC 进行扩展以解决多分类问题。

对 20 种不同的真实神经影像数据集进行的实验研究证实，所提出的方法的性能优于其它经典和前沿算法。除了论文中的理论和实证研究之外，我们还创建了一个基于图形用户界面（Graphical User Interface, GUI）的工具箱使得研究易于重现并向公众开放，该工具箱可以运行包括本文提出的方法在内的基于任务的 fMRI 图像分析方法的标准流程，目前该工具箱已在 `https://easyfmri.github.io` 上公开。此外，我们还提供了一个数据仓库用于共享本文所使用的数据集，该仓库位于 `https://easyfmridata.github.io`。

关键词：脑影像分析，深度监督功能校准，深度表征相似性分析，不平衡分类，集成学习。

ABSTRACT

One of the most significant challenges in both neuroscience and machine learning is comprehending how the human brain works. A better understanding of the brain–as the provenance of thoughts, memories, and emotions–expedites the development of society, including science, medicine, education, etc. As an imaging technology, functional Neuroimaging can be employed as a proxy for measuring the neural activation. The main idea is utilizing these measurements of neural activities to shed light on cognitive processes. Indeed, these images enable us to ask what information is represented in a region of the human brain and how that information is encoded, instead of asking what is a region's function. The neural activities can be analyzed at different levels, but a crucial step is knowing what the similarities (or differences) between distractive cognitive tasks are and then creating cognitive models to analyze the neural activities. It is like a spotlight that allows us to increase our knowledge related to the human brain and facilitate treatment of mental disease.

The past decade and a half have seen some promising advances in the development of approaches for decoding human neural activities. However, there are still several long-standing challenges, including functional alignment of multi-subject data, selecting information-rich features, similarity analysis by using unsupervised techniques and generating supervised models for predicting the neural activities. In this thesis, we development novel approaches in order to analyze the cognitive process in different levels and applications. The primary contributions can be summarized as follows:

(1) We introduce two techniques for functional alignment, i.e., unsupervised and supervised approaches. First, Deep Hyperalignment (DHA) is developed as a novel unsupervised approach for functional alignment. DHA employs a deep kernel function, including a multi-layer neural network, which can separately implement *any nonlinear function*. Furthermore, DHA uses rank-m SVD and Stochastic Gradient Descent (SGD) for optimization. Consequently, DHA generates low-runtime on large datasets, and the training data is not referenced when DHA computes the functional alignment for a new subject. As a supervised alternative, Local Discriminant Hyperalignment (LDHA) method is introduced by incorporating the idea of Local Discriminate Analysis (LDA) into CCA for improving the performance

of the hyperalignment solution. In fact, the idea of locality is defined based on the stimuli categories (class labels) in the train-set, where the proposed method firstly generates two sets for each category of stimuli, i.e., the set of nearest homogeneous stimuli as within-class neighborhoods and the set of stimuli from distinct categories as between-class neighborhoods. Then, these two sets are used to provide a better HA solution, where the correlation between the within-class neighborhoods is maximized, and also the correlation among between-class neighborhoods approaches to near zero. Finally, we illustrate that while DHA can improve the performance of binary analysis, LDHA depicts better performance for multi-class datasets.

(2) We propose new feature analysis techniques. Indeed, the proposed method selects a snapshot of brain image for each stimulus rather than analyzing whole of the time series. While the classical methods just can extract features from voxel space, the proposed method selects a subset of time-points for decoding the neural codes. In fact, these snapshots are selected by finding local maximums in the smoothed version of the design matrix. Finally, we propose two learning approaches. Indeed, extracted features can be analyzed by using both unsupervised learning and supervised learning. This thesis proposed a cluster ensemble approach in order to apply unsupervised learning, where similarities or distances between neural activities can be compared across subjects. As the supervised alternative, we develop a bagging technique by using binary $\ell1$-regularized SVM classifiers, where they are generated by utilizing each of neural activities in the level of anatomical regions. The main contribution of this method is that it can decrease the sparsity of fMRI datasets.

(3) We develop Deep Representational Similarity Analysis (DRSA) for similarity analysis. DRSA employs a deep kernel function, which transforms nonlinear neural activities into a linear embedded space and then evaluates the similarities (or distances) between the mapped features in that space. Moreover, it employs a new regularization term that can make a tread-off between correlation and covariance of different categories of stimuli. Since DRSA utilizes gradient-based optimization approaches, it is time efficient for evaluating high-dimensional Neuroimaging data, such as whole-brain images.

(4) We also introduce novel supervised learning techniques. First, Imbalance AdaBoost Binary Classification (IABC) is proposed as a novel technique for binary imbalance-classification

learning, where it is well-suited for one-vs-all classification analysis. Indeed, IABC uses a supervised random sampling and penalty values, which are calculated by the correlation between different classes, for improving the performance of prediction. After that, we extend IABC for the multi-class problems, by utilizing Error-Correcting Output Codes (ECOC) method.

Experimental studies on 20 different real-world Neuroimaging datasets confirm that the proposed methods achieve superior performance to other classical and state-of-the-art algorithms. Besides the theories and the empirical studies in the thesis, we also make our research easily reproducible and open to the public. We have created a GUI-based toolbox for running the standard pipeline of analyzing task-based fMRI images, including the proposed methods in this thesis, that is available at `https://easyfmri.github.io`. Moreover, we have also prepared a data repository for sharing employed datasets in this thesis. This repository is available at `https://easyfmridata.github.io`.

Keywords: Neuroimaging analysis, deep and supervised functional alignment, deep representational similarity analysis, imbalance classification, ensemble learning.

Contents

List of Figures

List of Tables

Notations

x	Scalar
$\mathbb{R}$	Set of real numbers
$\mathbb{Z}$	Set of integer numbers
$\mathbf{x}$	Vector
$\mathbf{x}^{(\ell)}$	Vector related to ℓ instance/session/subject
x_ℓ	ℓ-th element of vector $\mathbf{x}$
$\mathbf{X}$	Matrix
$\mathbf{I}$	Identity Matrix
$\mathbf{X}^{(\ell)}$	Matrix related to ℓ instance/session/subject
$\mathbf{X}_\ell$, $\mathbf{x}_{\ell.}$, $\mathbf{x}_{.\ell}$	ℓ-th row/column vector of matrix $\mathbf{X}$
$\mathbf{X}_{ij}$, $\mathbf{x}_{ij}$	Matrix cell located at i-th row and j-th column
$\{x_i\}$	Set with element x_i
$[X1, X2]$	Interval with lower bound $X1$ and upper bound $X2$
$f(\mathbf{X})$	Function on $\mathbf{X}$
$f(\mathbf{X}; \boldsymbol{\theta})$	Function on $\mathbf{X}$ with parameters $\boldsymbol{\theta}$
$\nabla\mathbf{X}$	The gradient of $\mathbf{X}$
$\partial\mathbf{X}$	The partial of $\mathbf{X}$
$\mathbb{E}[\mathbf{X}]$	Expected Value of $\mathbf{X}$

Abbreviations

Abbreviation	Full Name
AC	Abstract-Category
AdaBoost	Adapting Boosting
ANN	Artificial Neural Network
BOLD	Blood-Oxygen-Level-Dependent
BGCM	graph-based consensus maximization
BRSA	Bayesian RSA
CAE	Convolutional Autoencoder
CCA	Canonical Correlation Analysis
CR	Correlation Ratio
DHA	Deep Hyperalignment
DRSA	Deep Representational Similarity Analysis
DSM	Dissimilarity Matrix
ECOC	Error-Correcting Output Codes
ECoG	electrocorticography
EEG	electroencephalography
EROS	Event-Related Optical Signal
FFA	Fusiform Face Area
fMRI	functional Magnetic Resonance Imaging
GLM	General Linear Model
GRSA	Gradient RSA
HA	Hyperalignment
HRF	Hemodynamic Response Function
IABC	Imbalance AdaBoost Binary Classification
ICA	Independent Component Analysis

ISC	Inter-Subject Correlation
JE	Joint Entropy
KHA	Kernel Hyperalignment
PCA	Principal Component Analysis
PPA	Parahippocampal Place Area
LASSO	Least Absolute Shrinkage and Selection Operator
LDA	Linear Discriminant Analysis
LDHA	Local Discriminant Hyperalignment
MEG	Magnetoencephalography
MLP	Multilayer Perceptron
MNI	Montreal Neurological Institute
MREL	Multi-Region Ensemble Learning
MRI	Magnetic Resonance Imaging
MVP	Multivariate Pattern
MVPA	Multi-Voxel Pattern Analysis
MI	Mutual Information
NIRS	Near-Infrared Spectroscopy
NMI	Normalized Mutual Information
OLS	Ordinary Least Squares
OSCAR	Octagonal Shrinkage and Clustering Algorithm for Regression
OWL	Ordered Weighted $\ell 1$
PET	Positron Emission Tomography
ReLU	Rectified Linear Unit
SE	Specific-Exemplar
SGD	Stochastic Gradient Descent
SMI	Standardized Mutual Information
SNR	Signal-to-Noise Ratio

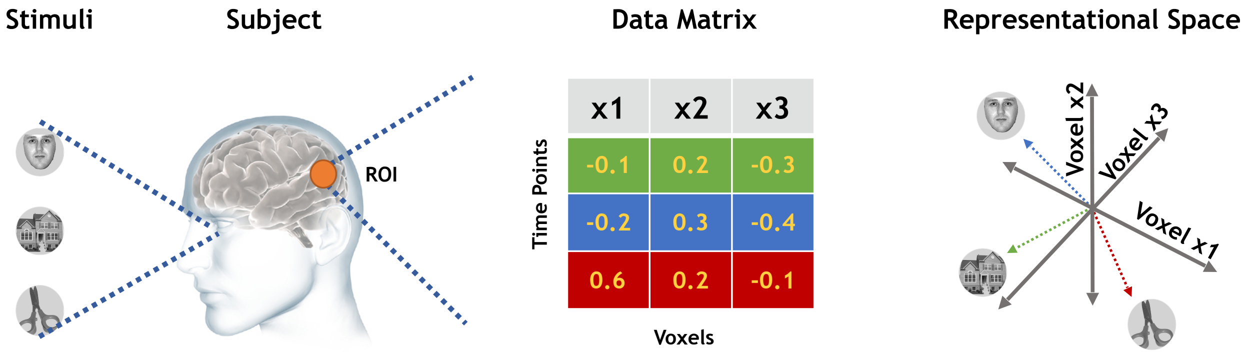
SPECT	Single-Photon Emission Computed Tomography
SVM	Support Vector Machines
SL	SearchLight
SLR	Sparse Logistic Regression
TKSC	Two Kernels Spectral Clustering
rs-fMRI	Rest-mode fMRI
RHA	Regularized Hyperalignment
ROI	Region of Interests
RSA	Representational Similarity Analysis
RSNs	resting-state networks
SMED	Stimulus-Model-based Encoding and Decoding
SVD	Singular Value Decomposition
SVDHA	Singular Value Decomposition Hyperalignment
SRM	Shared Response Model
WEAC	Weighted Evidence Accumulation Clustering
WSCE	Weighted Spectral Cluster Ensemble

Chapter 1. Introduction

One of the most significant challenges in our century is comprehending how the human brain works [1]. As an interdisciplinary field of study, computational neuroscience can break neural codes by employing different concepts from the mathematics, physics, psychology, psychiatry, and machine learning. In this thesis, we focus on developing modern machine learning approaches for analyzing the neural activities. We first present the current challenges for decoding the human brain in this section and then introduce novel techniques for improving this procedure in the rest of thesis.

1.1 How to measure the neural activities?

The neural activities can be analyzed at different levels, but a crucial step is knowing what the similarities (or differences) between distractive cognitive tasks are. In order to measure neural activities, different modalities of measurement can be utilized, including Event-Related Optical Signal (EROS), Positron Emission Tomography (PET), Single-Photon Emission Computed Tomography (SPECT), Near-Infrared Spectroscopy (NIRS), magnetoencephalography (MEG), electrocorticography (ECoG), electroencephalography (EEG), functional Magnetic Resonance Imaging (fMRI) [1, 2]. Like most of the previous studies [1–7], this thesis will focus on fMRI images, where this technique measures neural activity by employing the Blood-Oxygen-Level-Dependent (BOLD) contrast as a proxy for neural activation. The main idea is utilizing these measurements of neural activities to shed light on cognitive processes. Indeed, fMRI enables us to ask what information is represented in a region of the human brain and how that information is encoded, instead of asking what a region's function is [1]. There are two primary reasons for using fMRI technology to break the neural codes. Firstly, it is a non-invasive imaging technique [2, 5]. In comparison with other non-invasive brain imaging techniques, it also has unprecedented spatiotemporal resolution with no known side effects [1].



x1	x2	x3
-0.1	0.2	-0.3
-0.2	0.3	-0.4
0.6	0.2	-0.1

Figure 1.1 An example of representational space, where three visual stimuli are observed and depicted as three vectors with 3-dimensions.

1.2 Representational Space

Neural activities are analyzed in different forms, including graph structures, continuous signals, component-based representation, etc. In fMRI datasets, neural activities are usually in the form of voxels (volume elements in brain images). The core concept that underlies the human brain decoding is the high-dimensional and big data-representational space [4]. For instance, fMRI responses for a subject, including six time-points and 1000 voxels in the Region of Interests (ROIs), must be defined by a vector in a 6000-dimensional space. Indeed, the brain neural activities for each subject are considered as a vector in the neural representational space [1]. In other words, representational space for all subjects can be numerically defined by a matrix in which each column is a local pattern feature (i.e., all time points belong to a unique voxel) and each row denotes a response vector related to an individual simulation. Figure 1.1 illustrates an example of a representational space, where three visual stimuli are observed and depicted as three vectors with 3-dimensions. The primary advantage of using the concept of representational space is that we can generalize the applications of machine learning methods across different modalities of measurement [1, 2, 5].

1.3 Taxonomy of brain decoding methods

As Figure 1.2 depicted, brain patterns in the task-based fMRI datasets can be extracted and decoded by applying machine learning techniques, i.e., Hyperalignment (HA), Feature Analysis, Representational Similarity Analysis (RSA), Multivariate Pattern (MVP) classification, and Stimulus-Model-based Encoding and Decoding (SMED) [1, 4, 8, 9]. In this thesis, we consider that all datasets are preprocessed. We will explain different stages of preprocessing in the Appendix.

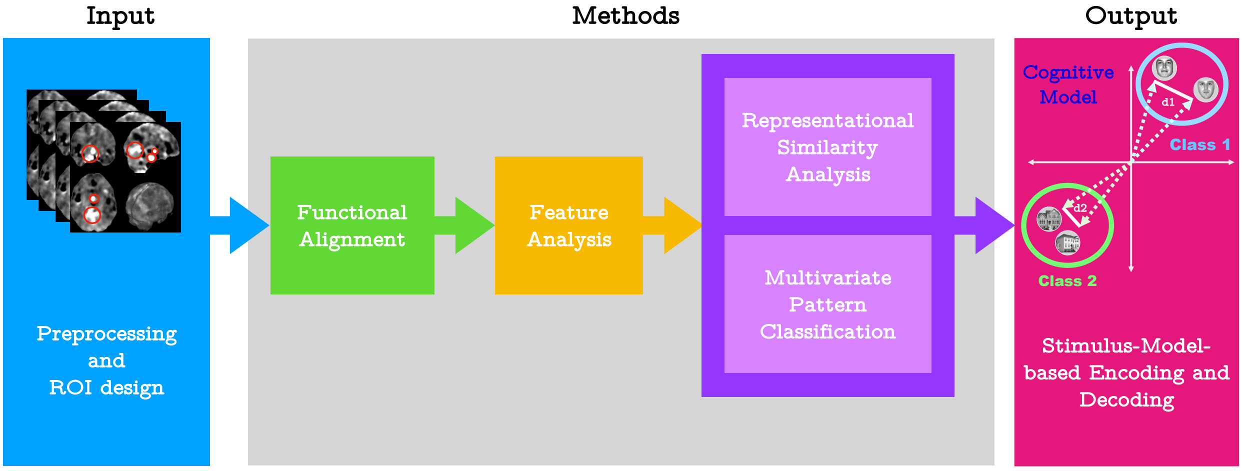

Input
Methods
Output
Preprocessing and ROI design
Functional Alignment
Feature Analysis
Representational Similarity Analysis
Multivariate Pattern Classification
Cognitive Model
d1
Class 1
d2
Class 2
Stimulus-Model-based Encoding and Decoding

Figure 1.2 Pipeline of analyzing neural activities

1.3.1 Functional Alignment

The multi-subject fMRI analysis is a challenging problem in the human brain decoding [4, 6, 9–14]. In fact, multi-subject fMRI images must be aligned across subjects in order to take between-subject variability into account. There are technically two main alignment methods, including anatomical alignment and functional alignment, which can work in unison. However, anatomical alignment can limitedly improve the accuracy because the size, shape and anatomical location of functional loci differ across subjects [6, 13–15]. As Figure 1.3 depicted, functional alignment explores to align the fMRI images across subjects precisely. Here, we are looking for a common /or shared space, where the correlation between within-class stimuli will be maximized, and the between-classes neural activities have significant distances in comparison with each other. In supervised learning, the shared space is generated by the training-set and then will be used for mapping neural activities in the testing-set [9].

As the widely used functional alignment method [4, 6, 9, 11–14], Hyperalignment (HA) [6] is an ‘anatomy free’ functional alignment method, which can be mathematically formulated as a multi-view representational learning problems [2, 4, 5, 11]. Original HA does not work in a very high dimensional space [9]. In order to extend HA into the real-world problems, Xu et al. developed the Regularized Hyperalignment (RHA) by utilizing an EM algorithm to seek the regularized optimum parameters iteratively [11]. Further, Chen et al. developed Singular Value Decomposition Hyperalignment (SVDHA), which firstly provides dimensionality reduction by SVD, and then HA aligns the functional responses in the reduced space [12]. In another study, Chen et al. introduced Shared

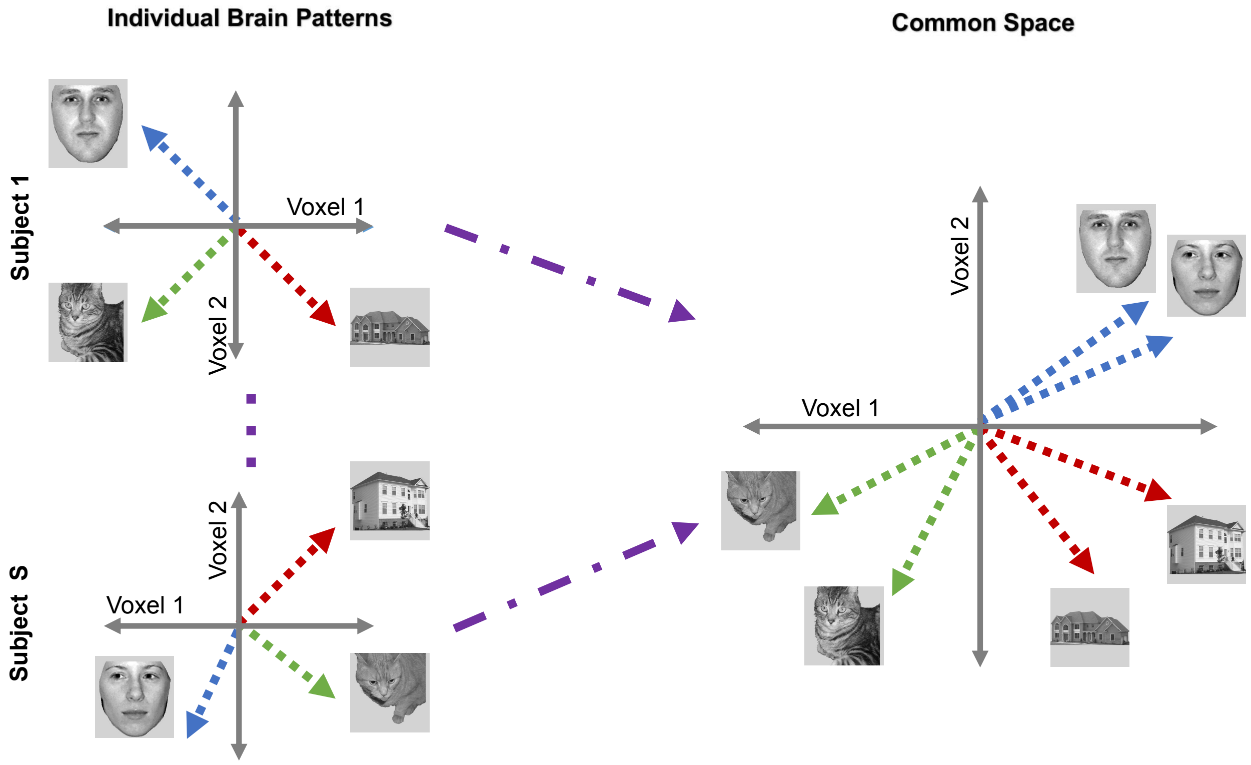


Figure 1.3 An example of functional alignment

Response Model (SRM), which is technically equivalent to Probabilistic CCA [9]. In addition, Guntupalli et al. developed SearchLight (SL) model, which is actually an ensemble of quasi-CCA models fits on patches of the brain images [15]. Lorbert et al. illustrated the limitation of HA methods on the linear representation of fMRI responses. They also proposed Kernel Hyperalignment (KHA) as a nonlinear alternative in an embedding space for solving the HA limitation [4]. Although KHA can solve the nonlinearity and high-dimensionality problems, its performance is limited by the fixed employed kernel function [13]. As another nonlinear HA method, Chen et al. recently developed Convolutional Autoencoder (CAE) for whole-brain functional alignment. Indeed, this method reformulates the SRM as a multi-view autoencoder [5] and then uses the standard SL analysis [15] in order to improve the stability and robustness of the generated classification (cognitive) model [16]. Since CAE simultaneously employs SRM and SL, its time complexity is so high [13]. In a nutshell, there are four main challenges in previous HA methods for calculating accurate functional alignments, i.e., nonlinearity [4, 13, 16], high-dimensionality [9, 12, 13], using a large number of subjects [13, 17], and supervised learning [14, 18].

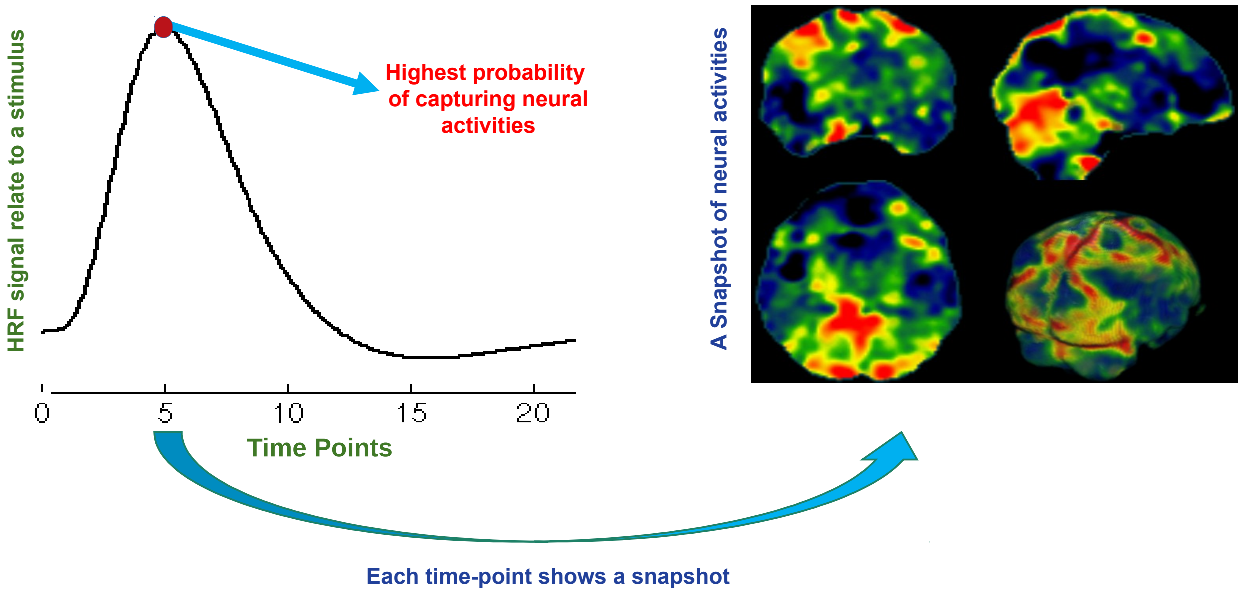


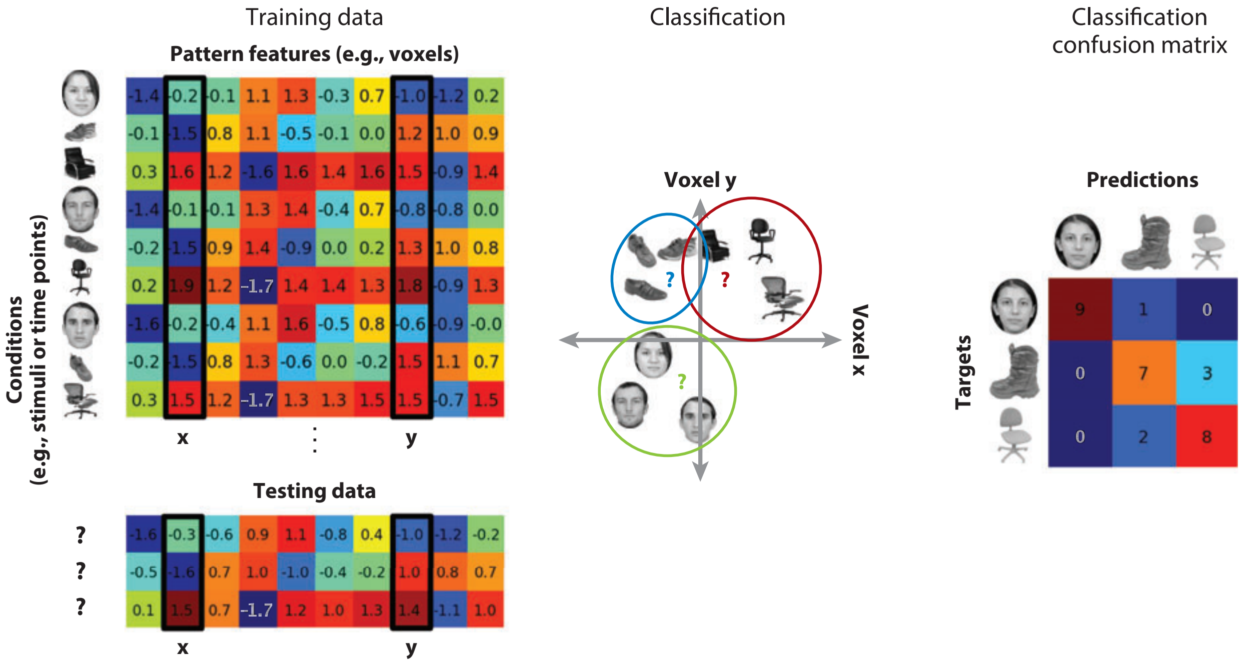
Figure 1.4 An example of time-point selection

1.3.2 Feature Analysis

Since most of fMRI datasets are high-dimensional, noisy, and sparse, some studies employed feature selection or extraction methods. Neural activities can be selected in two levels, i.e., voxels, or time points.

As the first group, some techniques project features of raw voxels to an embedded space, where the mapped features are information-rich, linear, etc. The component-based approaches such as Principal Component Analysis (PCA), Independent Component Analysis (ICA), or Linear Discriminant Analysis (LDA) are the most prevalent techniques in this stage [8, 19, 20]. As another alternative, SearchLight methods select features by analyzing the histogram of neuroimaging data [7, 15]. Indeed, there are two categories of SearchLight-based approaches, i.e., information-based technique, and spatial-resolution-based methods. While the information-based approach is only looking for the voxels with the highest-intensity, the spatial-resolution-based techniques consider the position of selected voxels [7]. Indeed, this thesis also presents a new approach for selecting features based on the anatomical structure of the human brain. While neural activities from any region (related, or irrelevant) are selected by using the component-based methods, the anatomical-based approach employs spatial information for selecting features.

The next group of methods focused on selecting a subset of time points rather than using the


Training data
Pattern features (e.g., voxels)
Conditions (e.g., stimuli or time points)
x
⋮
y
Testing data
?
?
?
x
y
Classification
Voxel y
Voxel x
?
?
?
Classification confusion matrix
Predictions
Targets

-1.4	-0.2	-0.1	1.1	1.3	-0.3	0.7	-1.0	-1.2	0.2
-0.1	-1.5	0.8	1.1	-0.5	-0.1	0.0	1.2	1.0	0.9
0.3	1.6	1.2	-1.6	1.6	1.4	1.6	1.5	-0.9	1.4
-1.4	-0.1	-0.1	1.3	1.4	-0.4	0.7	-0.8	-0.8	0.0
-0.2	-1.5	0.9	1.4	-0.9	0.0	0.2	1.3	1.0	0.8
0.2	1.9	1.2	-1.7	1.4	1.4	1.3	1.8	-0.9	1.3
-1.6	-0.2	-0.4	1.1	1.6	-0.5	0.8	-0.6	-0.9	-0.0
-0.2	-1.5	0.8	1.3	-0.6	0.0	-0.2	1.5	1.1	0.7
0.3	1.5	1.2	-1.7	1.3	1.3	1.5	1.5	-0.7	1.5

-1.6	-0.3	-0.6	0.9	1.1	-0.8	0.4	-1.0	-1.2	-0.2
-0.5	-1.6	0.7	1.0	-1.0	-0.4	-0.2	1.0	0.8	0.7
0.1	1.5	0.7	-1.7	1.2	1.0	1.3	1.4	-1.1	1.0

9	1	0
0	7	3
0	2	8

Figure 1.5 An example of Multivariate Pattern (MVP) classification [1]

whole of time series. Pioneer studies employed the component-based approaches such as PCA [21] and ICA [22] to select the subset of time points. In this thesis, we also present a new approach by using the design matrix for selecting time points with the highest probability of visualizing neural activities of different stimuli. As Figure 1.4 illustrates, we analyze the Hemodynamic Response Function (HRF) signal related to each stimulus and then select the time point in the local maximum, where this time point includes a snapshot of the neuroimaging data with the highest probability of demonstrating efficient neural activities corresponding to that stimulus.

1.3.3 Multivariate Pattern (MVP)

Multivariate Pattern (MVP) classification is a conjunction between neuroscience and computer science, which can extract and decode brain patterns by applying the classification methods [23]. In fact, it can predict patterns of neural activities associated with different cognitive states [24, 25] and also can define decision surfaces to distinguish different stimuli for decoding the brain and understanding how it works [1, 26]. Further, MVP classification can enable us to understand how brain stores and processes distinctive stimuli. Moreover, MVP classification uses machine learning algorithms to classify response patterns, associating each neural response with an experimental condition.

Pattern classification involves defining sectors in the neural representational space in which all response vectors represent the same class of information, such as a stimulus category [1], an attended stimulus [8], or a cognitive state [24, 25]. It can also be used to find novel treatments for mental diseases or even to create a new generation of the user interface.

As Figure 1.5 depicted, an MVP classification analysis firstly divides the data into independent training and testing datasets. Then, the decision rules that determine the confines of each class of neural response vectors are developed on the training data. The border between sectors for different conditions is called a decision surface. The validity of the classifier is then tested on the independent test data. For valid generalization testing, the test data must play no role in the development of the classifier, including data preprocessing. Each test data response vector is then classified as another exemplar of the condition associated with the sector in which it is located [1].

Classifier accuracy is the percentage of test vectors that are correctly classified. A more revealing assessment of classifier performance is afforded by examining the confusion matrix. A confusion matrix presents the frequencies for all classifications of each experimental condition, including the details about misclassifications. Examination of misclassifications adds information about which conditions are most distinct and which are more similar. This information is analyzed using additional methods in RSA.

1.3.4 Representational Similarity Analysis (RSA)

As one of the fundamental approaches in fMRI analysis, Representational Similarity Analysis (RSA) [27] evaluates the similarities (or distances) between distractive cognitive tasks. As Figure 1.6 illustrated, RSA examines the structure of representations within a representational space regarding distances between response vectors. Such as clustering analysis in machine learning, the complete set of distances among all pairs of response vectors is known as the dissimilarity matrix (DSM). Figure 1.7 shows an example of DSM matrix. Whereas MVP classification analyzes whether the vectors for different conditions are clearly distinct, RSA analyzes how they are related to each other. This approach confers several advantages. First, RSA can reveal that representations in different brain areas differ even if MVP classification is equivalent in those areas [27, 28]. Second, by converting the locations of response vectors from a set of feature coordinates to a set of distances between vectors, the geometry of the representational space is now in a format that is not dependent on the feature

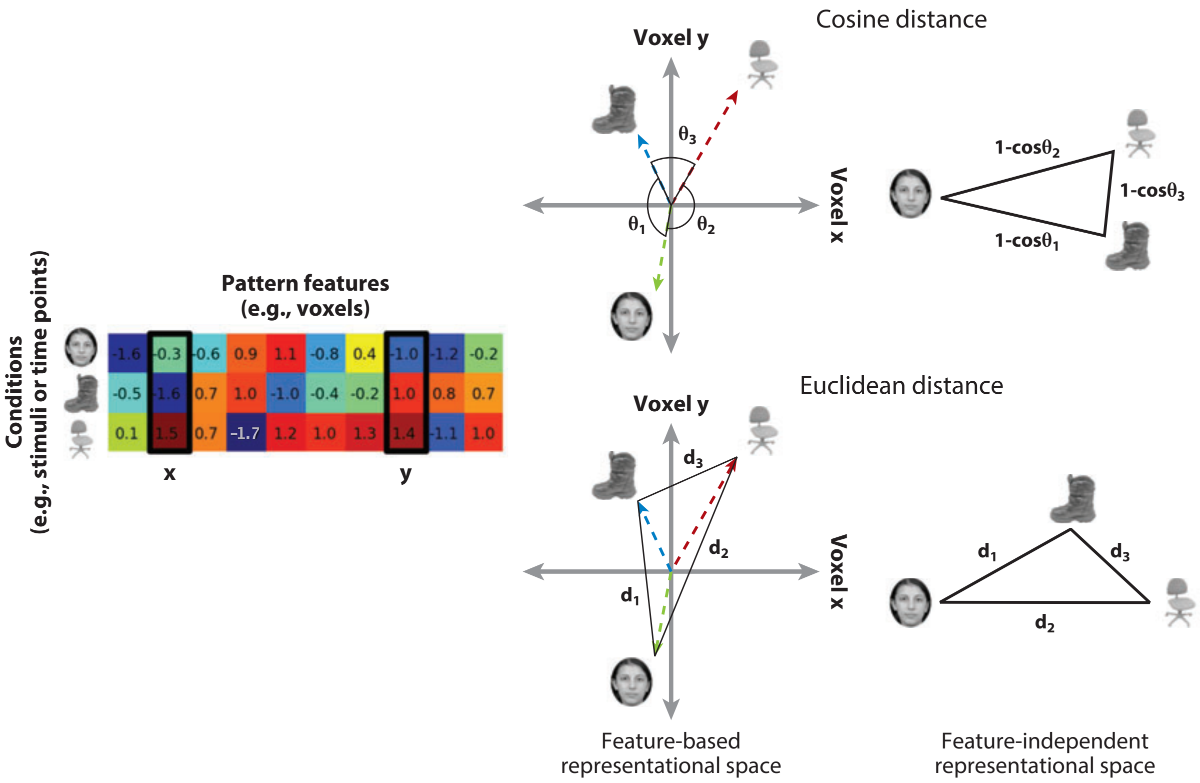


Figure 1.6 An example of Representational Similarity Analysis (RSA) [1]

coordinate axes [1, 27, 28]. Third, RSA can compare the neural activities across distinctive species. For instance, Figure 1.8 illustrates the comparisons of different visual stimuli between monkeys brain and human brains [28]. The main disadvantage of RSA techniques is that they do not create any cognitive models and the whole of RSA procedure must be repeated for analyzing a new subject. Consequently, RSA techniques are not computationally efficient [1].

In practice, RSA can be mathematically formulated as a multi-set (group) regression problem, i.e., a linear model for mapping between the matrix of neural activities and the design matrix. Original RSA employs basic linear approaches, such as Ordinary Least Squares (OLS) [7] or General Linear Model (GLM) [28]. Further, some of the modern approaches utilize the Bayesian technique [3, 29]. For instance, Bayesian RSA (BRSA) [3] considers the covariance matrix as a hyper-parameter generative model and then calculates this matrix from neural activities. As other alternatives, some studies employ regularized techniques, i.e., Ridge Regression [30], Least Absolute Shrinkage and Selection Operator (LASSO) [31], Elastic Net method [32], Octagonal Shrinkage and Clustering Algorithm for Regression (OSCAR) [33], and Ordered Weighted $\ell 1$ (OWL) [34, 35]. In some sense, RSA is similar to clustering techniques in machine learning [36–38].

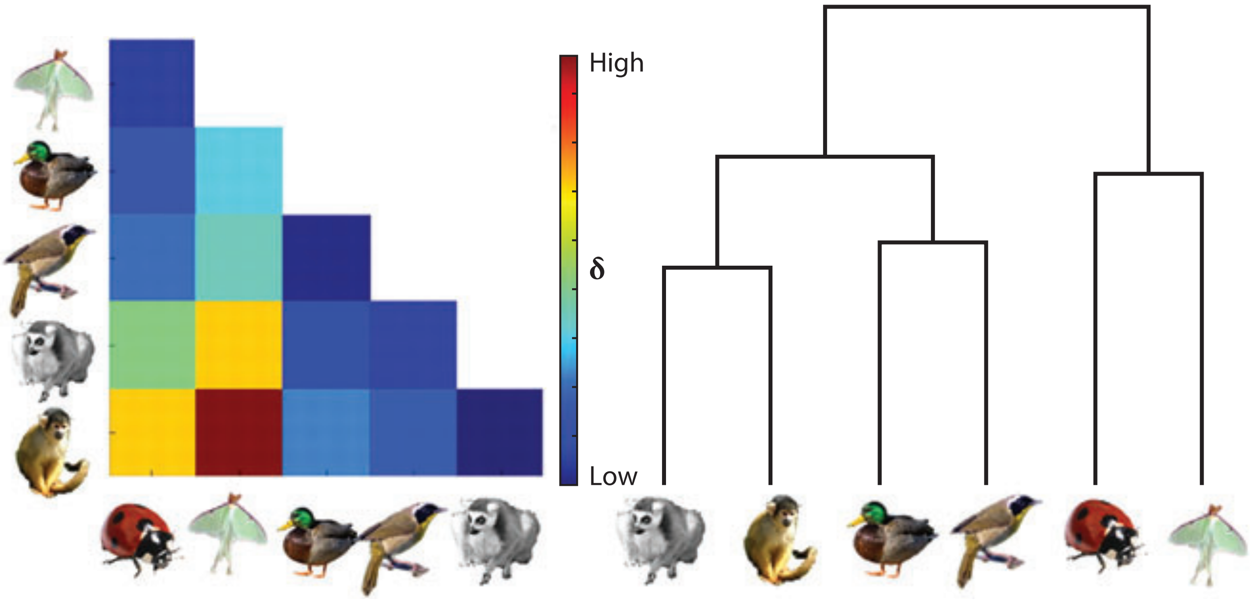


Figure 1.7 An example of Dissimilarity Matrix (DSM) [1, 27]

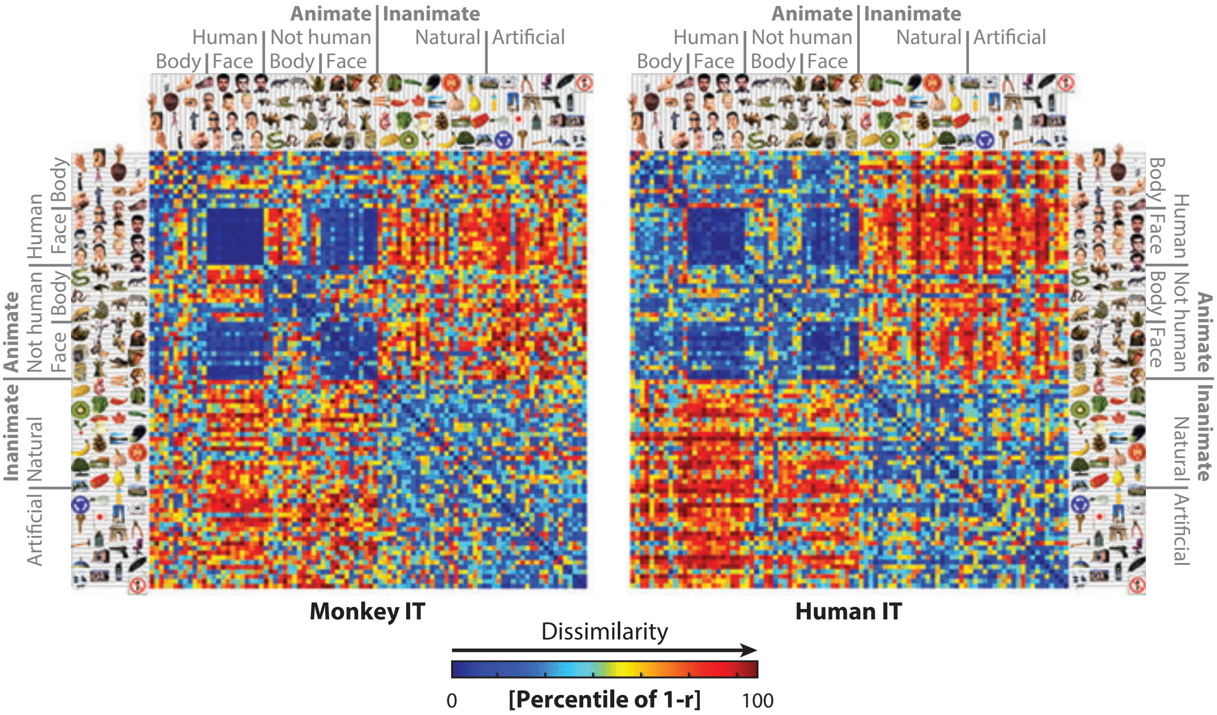


Figure 1.8 Comparing different visual stimuli across Monkey and Human brains [28]

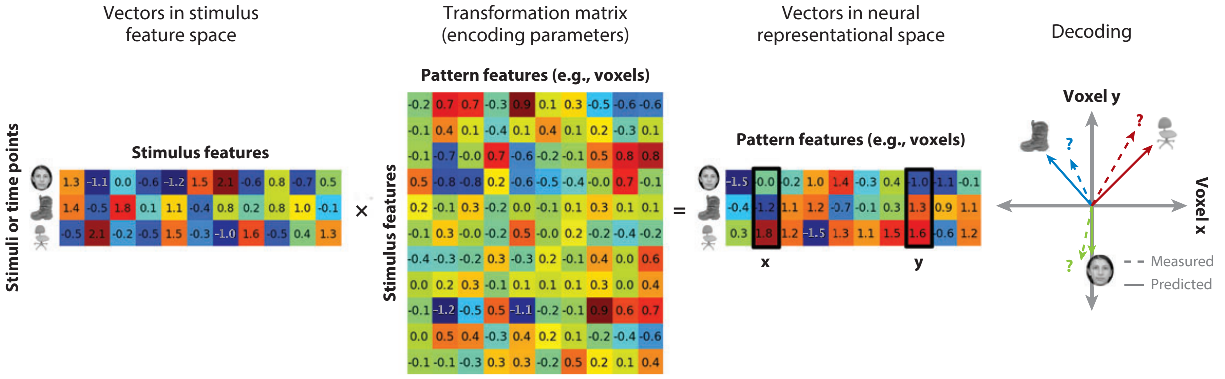

Vectors in stimulus feature space
Transformation matrix (encoding parameters)
Vectors in neural representational space
Decoding
Stimuli or time points
Stimulus features
×
Pattern features (e.g., voxels)
Stimulus features
=
Pattern features (e.g., voxels)
x
y
Voxel y
Voxel x
?
Measured
Predicted

Figure 1.9 An example of Stimulus-Model-based Encoding and Decoding (SMED) [1]

1.3.5 Stimulus-Model-based Encoding and Decoding (SMED)

The previous approaches, including MVP classification, RSA, and hyperalignment, compare different neural activities generated by distinctive stimuli in the same subject or across different subjects. Indeed, these methods cannot enable us to predict the neural activities for a novel stimulus or even a new dataset [1]. Stimulus-Model-based Encoding and Decoding (SMED) methods can extend the generated MVP /or RSA analysis for decoding the neural activities of the novel stimuli [39]. As Figure 1.9 depicted, SMED predicts the response to stimulus features rather than to the whole stimuli. Further, SMED technique focused on mapping vectors from feature space belonging to each stimulus to the shared neural spaces. Indeed, SMED is limited to Bayesian reconstruction based on similarities to a set of priors [1]. In practice, this technique can be useful for neuroscientist after a robust cognitive model is generated [1].

1.4 Organization of this thesis

In Chapter 2, we introduce two techniques for functional alignment. As the first technique, Local Discriminant Hyperalignment (LDHA) method incorporates the idea of Local Discriminate Analysis (LDA) into CCA [40] in order to improve the performance of the hyperalignment solution. Indeed, LDHA maximizes the correlation between homogeneous stimuli (from the same category) *and also removes the correlation between different categories of stimuli*. Furthermore, we propose Deep Hyperalignment (DHA) as a novel nonlinear approach for functional alignment. DHA is not limited by a restricted fixed representational space because the kernel in DHA is a multi-layer neural network, which can separately implement *any nonlinear function* [41, 42] for each subject to transfer the brain activities to the shared space.

In Chapter 3, we develop novel feature analysis techniques. While the classical methods just can extract features from voxel space, the proposed method discusses how to select a subset of time-points for analyzing the neural activities. Indeed, the proposed method estimates and analyzes a snapshot of brain image for each stimulus when the level of using oxygen is maximized. In other words, these snapshots are selected by finding local maximums in the smoothed version of the design matrix. The main advantage of this method is that it can decrease the sparsity of fMRI datasets. Further, we propose two learning approaches, including both unsupervised learning and supervised learning.

In Chapter 4, we propose Deep Representational Similarity Analysis (DRSA) as a new deep extension of RSA method for similarity analysis. Indeed, DRSA utilizes a deep network–including multiple stacked layers of nonlinear transformation–as the kernel function, which maps nonlinear neural activities to a linear information-rich embedded space and then evaluates the similarities (or distances) between the mapped features. Further, the proposed method employs a new regularization term that can make a trade-off between correlation and covariance of distinctive cognitive tasks. Since DRSA uses gradient-based optimization approaches, it is time efficient for evaluating high-dimensional fMRI images, such as whole-brain datasets.

In Chapter 5, we propose novel supervised learning techniques. Firstly, we introduce a novel approach for imbalance-classification learning. Indeed, this method is well-suited for one-vs-all classification analysis. Then, we focus on multi-class learning approach. Here, we utilize Error-Correcting Output Codes (ECOC) as an indirect multi-class approach in order to extend the proposed binary classifiers for the multi-class prediction.

In Chapter 6, we then present contributions and the conclusions. Besides the theories and the empirical studies in the thesis, we also make our research easily reproducible and open to the public. We have created a GUI-based toolbox for running the standard pipeline of analyzing task-based fMRI images, including the proposed methods in this thesis, that is available at `https://easyfmri.github.io`. Moreover, we have also prepared a data repository for sharing task-based fMRI datasets. This repository is available at `https://easyfmridata.github.io`. It includes more than 35 preprocessed real-world datasets related to distinctive cognitive tasks.

Figure 1.10 illustrates the graphical abstract of this thesis across chapters.

Prior Publications Parts of this thesis have been published in [8, 13, 14, 19, 20, 36–38].

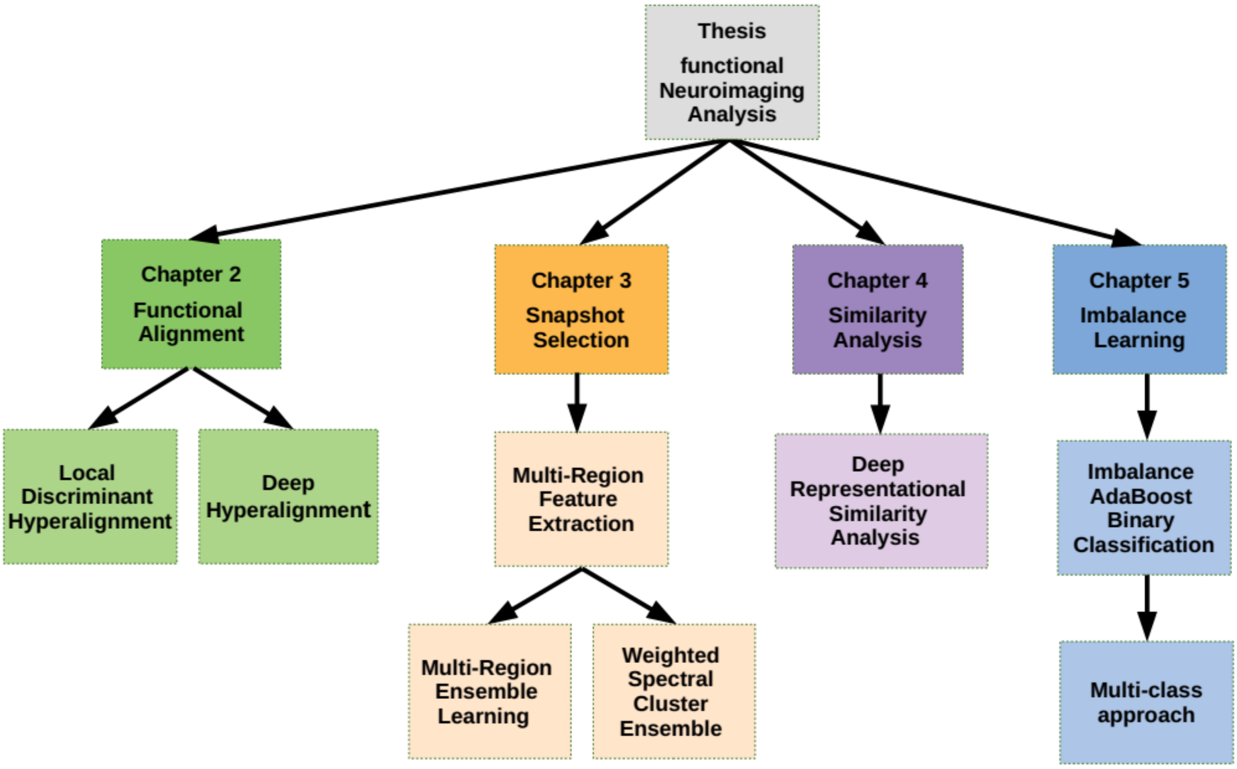


Figure 1.10 The graphical map of this thesis

Chapter 2. Supervised and Deep Hyperalignment

One of the main challenges in fMRI studies, especially MVP analysis, is using multi-subject datasets. On the one hand, the multi-subject analysis is critical to figure out the generality and validity of the generated results across subjects. On the other hand, analyzing multi-subject fMRI data requires accurate functional and anatomical alignments between neuronal activities of different subjects in order to increase the performance of the final results [6, 12]. Indeed, the fMRI datasets must be aligned across subjects in multi-subject studies in order to take between-subject variability into account. There are two main alignment approaches, i.e. anatomical alignment and functional alignment, which can work in unison. The anatomical alignment is the most common method for aligning fMRI images based on anatomical features by employing structural MRI images, e.g., Talairach alignment [43], or Montreal Neurological Institute (MNI) [44, 45]. However, this method generated limited accuracy since the size, shape and anatomical location of functional loci differ across subjects [46, 47]. Indeed, anatomical alignment is just used in many fMRI studies as a preprocessing step. By contrast, functional alignment seeks to directly align the brain neural responses across subjects.

2.1 Functional Alignment Techniques

There are several non-CCA based studies, which used functional and anatomical features for fMRI alignment. Conroy et al. introduced a new way to maximize the alignment of intra-subject patterns by utilizing the cortical warping [10]. Sabuncu et al. used cortical warping for maximizing the Inter-Subject Correlation (ISC) across subjects [48]. Dmochowski et al. maximized ISC by aggregating the subjects' data into an individual matrix [49]. Micheal et al. also proposed the GICA, IVA algorithms for rest-mode fMRI (rs-fMRI) functional alignment. Since this method does not assume time-synchronized stimulus, it concatenates data along the time dimension (implying spatial consistency) and learns spatial independent components [50].

As the widely used functional alignment method [4–6, 9, 11–14, 16], Hyperalignment (HA) [6] is an 'anatomy free' functional alignment method, which can be mathematically formulated as a multiple-set Canonical Correlation Analysis (CCA) problem [4, 11, 51]. Original HA does not work in a very high dimensional space [9, 13]. In order to extend HA into the real-world problems, Xu et al. developed the Regularized Hyperalignment (RHA) by utilizing an EM algorithm to iteratively seek the regularized optimum parameters [11]. Lorbert et al. illustrated the limitation of HA methods on the linear representation of fMRI responses. They also proposed Kernel Hyperalignment (KHA) as a nonlinear alternative in an embedding space for solving the HA limitation [4, 5]. Although KHA can solve the nonlinearity and high-dimensionality problems, its performance is limited by the fixed employed kernel function. Sui et al. employed the multimodal CCA and ICA approaches on multimodal data for identifying the unique and shared variance associated across modalities [52, 53]. Further, Chen et al. developed Singular Value Decomposition Hyperalignment (SVDHA), which firstly provides dimensionality reduction by SVD, and then HA aligns the functional responses in the reduced space [12]. In another study, Chen et al. introduced Shared Response Model (SRM), which is technically equivalent to Probabilistic CCA [16].

Guntupalli et al. developed a linear model of shared representational spaces based on the original HA. This model can trace fine-scale distinctions among varied responses with response-tuning basis functions that are common across subjects and models. Indeed, this method is actually an ensemble of quasi-CCA models fits on patches of the brain images [15]. As another nonlinear HA method, Chen et al. recently developed Convolutional Autoencoder (CAE) for *whole brain* functional alignment. Indeed, this method reformulates the SRM as a multi-view autoencoder and then uses the standard SL analysis [15] in order to improve the stability and robustness of the generated classification (cognitive) model. Since CAE simultaneously employs SRM and SL, its time complexity is so high. Turek et al. proposed a semi-supervised HA that simultaneously applies the alignment and performs the analysis. Indeed, this method also used the SRM [9] for alignment and then Multinomial Logistic Regression is employed for classification [18].

2.2 Hyperalignment

Preprocessed fMRI time series collected for S subjects can be defined by $\mathbf{X}^{(i)} = \left\{x_{mn}^{(i)}\right\} \in \mathbb{R}^{T\times V}, i = 1{:}S, m = 1{:}T, n = 1{:}V$, where T denotes the number of time points in unites of TRs (Time of Repetition), V is the number of voxels, and $x_{mn}^{(i)} \in \mathbb{R}$ denotes the functional activity for the $i\text{-}th$ subject in the $m\text{-}th$ time point and the $n\text{-}th$ voxel. For simplicity, assume that all data points are normalized by zero-mean and unit-variance, $\mathbf{X}^{(i)} \sim \mathcal{N}(0,1), i = 1{:}S$. If the original data points are not normalized, we can consider this assumption as a preprocessing step. Since there are more voxels than TRs in most of the fMRI studies, $\mathbf{X}^{(i)}$ and the voxel correlation map $(\mathbf{X}^{(i)})^\top\mathbf{X}^{(j)}$ may not be full rank [4, 5, 9, 11–14, 16]. In training-set, time synchronized stimulus ensures temporal alignment, i.e. the $m\text{-}th$ time point for all of the subjects represents the same simulation [11, 12]. Indeed, the main goal of HA methods is aligning the columns of $\mathbf{X}^{(i)}$ across subjects [4, 14]. In previous studies, Inter-Subject Correlation (ISC) was defined for functional alignment between two distinct subjects as follows, where $tr()$ is the trace function: [4, 5, 11, 12]

$$\text{ISC}(\mathbf{X}^{(i)}, \mathbf{X}^{(j)}) = ({}^{1}\!/\!{}_{V})\text{tr}\Big((\mathbf{X}^{(i)})^\top\mathbf{X}^{(j)}\Big). \tag{2.1}$$

By considering $\mathbf{X}^{(\ell)} \sim \mathcal{N}(0,1)$, $\ell = 1{:}S$ as column-wise standardized, ISC lies in $[-1,+1]$. Here, the large values illustrate better alignment [4, 11]. Based on (2.1), Hyperalignment (HA) problem can be defined as follows: [4, 5, 9, 11, 12]

$$\max_{\mathbf{R}^{(i)},\mathbf{R}^{(j)}} \sum_{i=1}^{S}\sum_{j=i+1}^{S} \text{ISC}(\mathbf{X}^{(i)}\mathbf{R}^{(i)}, \mathbf{X}^{(j)}\mathbf{R}^{(j)}) \qquad \text{s.t.} \quad \big(\mathbf{R}^{(\ell)}\big)^\top\widetilde{\mathbf{\Phi}}^{(\ell)}\mathbf{R}^{(\ell)} = \mathbf{I}, \ell = 1{:}S, \tag{2.2}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{R}^{(\ell)} = \Big\{r_{mn}^{(\ell)}\Big\} \in \mathbb{R}^{V\times V}$ denotes the solution for $\ell\text{-}th$ subject, and the matrices $\widetilde{\mathbf{\Phi}}^{(\ell)} \in \mathbb{R}^{V\times V}, \ell = 1{:}S$ are symmetric and positive definite. By considering $\widetilde{\mathbf{\Phi}}^{(\ell)} = \mathbf{I}$, (2.2) is equivalent to a multi-set orthogonal Procrustes problem [51], which is commonly used in share analysis. Furthermore, if $\widetilde{\mathbf{\Phi}}^{(\ell)} = (\mathbf{X}^{(\ell)})^\top\mathbf{X}^{(\ell)}$, then (2.2) denotes a form of multi-set Canonical Correlation Analysis (CCA) [4, 11] as follows:

$$\max_{\mathbf{R}^{(i)},\mathbf{R}^{(j)}} \sum_{i=1}^{S}\sum_{j=i+1}^{S} \text{tr}\Big((\mathbf{X}^{(i)}\mathbf{R}^{(i)})^\top\mathbf{X}^{(j)}\mathbf{R}^{(j)}\Big) \quad \text{s.t.} \quad \big(\mathbf{X}^{(\ell)}\mathbf{R}^{(\ell)}\big)^\top\mathbf{X}^{(\ell)}\mathbf{R}^{(\ell)} = \mathbf{I}, \quad \ell = 1{:}S. \tag{2.3}$$

Lemma 2.1. (2.3) *maximizes the Pearson correlation (cov$(A,B)/\sigma_A\sigma_B$) between each pair of mapped features ($\mathbf{X}^{(\ell)}\mathbf{R}^{(\ell)}$) across subjects [11, 40].*

Proof. By considering $\widetilde{\mathbf{\Phi}}^{(\ell)} \in \mathbb{R}^{V\times V} = \mathbb{E}\Big[(\mathbf{X}^{(\ell)})^\top\mathbf{X}^{(\ell)}\Big] = (\mathbf{X}^{(\ell)})^\top\mathbf{X}^{(\ell)}$ as Expectation for population, the Pearson correlation on (2.3) can be defined as follows:

$$\max_{\mathbf{R}^{(i)},\mathbf{R}^{(j)}} \sum_{i=1}^{S}\sum_{j=i+1}^{S} \text{tr}\Bigg(\frac{(\mathbf{X}^{(i)}\mathbf{R}^{(i)})^\top\mathbf{X}^{(i)}\mathbf{R}^{(j)}}{\sqrt{((\mathbf{R}^{(i)})^\top\widetilde{\mathbf{\Phi}}^{(i)}\mathbf{R}^{(i)})}\sqrt{((\mathbf{R}^{(j)})^\top\widetilde{\mathbf{\Phi}}^{(j)}\mathbf{R}^{(j)})}}\Bigg). \tag{2.4}$$

Since $\big(\mathbf{R}^{(\ell)}\big)^\top\widetilde{\mathbf{\Phi}}^{(\ell)}\mathbf{R}^{(\ell)} = \mathbf{I}$, we approach from (2.4) to (2.3). □

The constrains must be imposed in $\mathbf{R}^{(\ell)}$ to avoid overfitting [11]. In order to seek an optimum solution, solving (2.3) may not be the best approach because there is no scale to evaluate the distance between current result and the optimum (fully maximized) solution [4, 13].

Lemma 2.2. (2.3) *can be rewritten as following minimization problem:*

$$\min_{\mathbf{R}^{(i)},\mathbf{R}^{(j)}} \sum_{i=1}^{S} \sum_{j=i+1}^{S} \left\| \mathbf{X}^{(i)}\mathbf{R}^{(i)} - \mathbf{X}^{(j)}\mathbf{R}^{(j)} \right\|_F^2, \quad \text{s.t.} \quad \left(\mathbf{X}^{(\ell)}\mathbf{R}^{(\ell)}\right)^\top \mathbf{X}^{(\ell)}\mathbf{R}^{(\ell)} = \mathbf{I}, \quad \ell = 1{:}S, \tag{2.5}$$

where (2.5) *approaches zero for an optimum result.*

Proof.

$$\min_{\mathbf{R}^{(i)},\mathbf{R}^{(j)}} \sum_{i=1}^{S} \sum_{j=i+1}^{S} \left\| \mathbf{X}^{(i)}\mathbf{R}^{(i)} - \mathbf{X}^{(j)}\mathbf{R}^{(j)} \right\|_F^2 = \min_{\mathbf{R}^{(i)},\mathbf{R}^{(j)}} \sum_{i=1}^{S} \sum_{j=i+1}^{S} \Bigg(\text{tr}\Big(\big(\mathbf{X}^{(i)}\mathbf{R}^{(i)}\big)^\top \mathbf{X}^{(i)}\mathbf{R}^{(i)} \Big) +$$
$$\text{tr}\Big(\big(\mathbf{X}^{(j)}\mathbf{R}^{(j)}\big)^\top \mathbf{X}^{(j)}\mathbf{R}^{(j)} \Big) - 2\text{tr}\Big(\big(\mathbf{X}^{(i)}\mathbf{R}^{(i)}\big)^\top \mathbf{X}^{(j)}\mathbf{R}^{(j)} \Big) \Bigg).$$

Since $\left(\mathbf{X}^{(\ell)}\mathbf{R}^{(\ell)}\right)^\top \mathbf{X}^{(\ell)}\mathbf{R}^{(\ell)} = \mathbf{I}$, so we have:

$$\min_{\mathbf{R}^{(i)},\mathbf{R}^{(j)}} \sum_{i=1}^{S} \sum_{j=i+1}^{S} \Bigg(2V_{new} - 2\text{tr}\Big(\big(\mathbf{X}^{(i)}\mathbf{R}^{(i)}\big)^\top \mathbf{X}^{(j)}\mathbf{R}^{(j)} \Big) \Bigg) \equiv \max_{\mathbf{R}^{(i)},\mathbf{R}^{(j)}} \sum_{i=1}^{S} \sum_{j=i+1}^{S} \text{tr}\Big((\mathbf{X}^{(i)}\mathbf{R}^{(i)})^\top \mathbf{X}^{(j)}\mathbf{R}^{(j)} \Big).$$

□

The main assumption in the original HA is that the $\mathbf{R}^{(\ell)}, \ell = 1{:}S$ are noisy 'rotations' of a shared space (or common template) [6].

Lemma 2.3. *The equation* (2.5) *is equivalent to:*

$$\min_{\mathbf{R}^{(i)},\mathbf{G}} \sum_{i=1}^{S} \left\| \mathbf{X}^{(i)}\mathbf{R}^{(i)} - \mathbf{G} \right\|_F^2, \qquad \text{s.t.} \quad \left(\mathbf{X}^{(\ell)}\mathbf{R}^{(\ell)}\right)^\top \mathbf{X}^{(\ell)}\mathbf{R}^{(\ell)} = \mathbf{I}, \quad \ell = 1{:}S, \tag{2.6}$$

where $\mathbf{G} \in \mathbb{R}^{T \times V}$ *is the shared space:*

$$\mathbf{G} = \frac{1}{S} \sum_{j=1}^{S} \mathbf{X}^{(j)}\mathbf{R}^{(j)} \tag{2.7}$$

Proof.

$$\min_{\mathbf{R}^{(i)},\mathbf{R}^{(j)}} \sum_{i=1}^{S} \sum_{j=i+1}^{S} \left\| \mathbf{X}^{(i)}\mathbf{R}^{(i)} - \mathbf{X}^{(j)}\mathbf{R}^{(j)} \right\|_F^2 =$$

$$\min_{\mathbf{R}^{(i)},\mathbf{R}^{(j)}} \sum_{i=1}^{S} \sum_{j=i+1}^{S} \left(\mathrm{tr}\Big(\big(\mathbf{X}^{(i)}\mathbf{R}^{(i)}\big)^{\top} \mathbf{X}^{(i)}\mathbf{R}^{(i)} \Big) + \mathrm{tr}\Big(\big(\mathbf{X}^{(j)}\mathbf{R}^{(j)}\big)^{\top} \mathbf{X}^{(j)}\mathbf{R}^{(j)} \Big) - 2\mathrm{tr}\Big(\big(\mathbf{X}^{(i)}\mathbf{R}^{(i)}\big)^{\top} \mathbf{X}^{(j)}\mathbf{R}^{(j)} \Big) \right)$$

$$\equiv \min_{\substack{\mathbf{R}^{(i)},\mathbf{R}^{(j)},\\ \mathbf{G}}} \frac{1}{2} \sum_{i=1}^{S} \left(S\mathrm{tr}\Big(\big(\mathbf{X}^{(i)}\mathbf{R}^{(i)}\big)^{\top} \mathbf{X}^{(i)}\mathbf{R}^{(i)} \Big) - 2S\mathrm{tr}\Big(\big(\mathbf{X}^{(i)}\mathbf{R}^{(i)}\big)^{\top} \mathbf{G} \Big) + \sum_{j=1}^{S} \left(\mathrm{tr}\Big(\big(\mathbf{X}^{(j)}\mathbf{R}^{(j)}\big)^{\top} \mathbf{X}^{(j)}\mathbf{R}^{(j)} \Big) \right) \right)$$

$$= \min_{\substack{\mathbf{R}^{(i)},\mathbf{R}^{(j)},\\ \mathbf{G}}} \frac{1}{2} \left(\left(S \sum_{i=1}^{S} \mathrm{tr}\Big(\big(\mathbf{X}^{(i)}\mathbf{R}^{(i)}\big)^{\top} \mathbf{X}^{(i)}\mathbf{R}^{(i)} \Big) \right) - \Big(2S^2 \mathrm{tr}\big(\mathbf{G}^{\top}\mathbf{G}\big) \Big) + \left(S \sum_{j=1}^{S} \mathrm{tr}\Big(\big(\mathbf{X}^{(j)}\mathbf{R}^{(j)}\big)^{\top} \mathbf{Y}^{(j)}\mathbf{R}^{(j)} \Big) \right) \right)$$

$$= \min_{\mathbf{R}^{(i)},\mathbf{G}} \frac{1}{2} \left(\left(2S \sum_{i=1}^{S} \mathrm{tr}\Big(\big(\mathbf{X}^{(i)}\mathbf{R}^{(i)}\big)^{\top} \mathbf{X}^{(i)}\mathbf{R}^{(i)} \Big) - \Big(2S^2 \mathrm{tr}\big(\mathbf{G}^{\top}\mathbf{G}\big) \Big) \right) \right)$$

$$= \min_{\mathbf{R}^{(i)},\mathbf{G}} \left(\left(S \sum_{i=1}^{S} \mathrm{tr}\Big(\big(\mathbf{X}^{(i)}\mathbf{R}^{(i)}\big)^{\top} \mathbf{X}^{(i)}\mathbf{R}^{(i)} \Big) \right) - \Big(S^2 \mathrm{tr}\big(\mathbf{G}^{\top}\mathbf{G}\big) \Big) \right)$$

$$= \min_{\mathbf{R}^{(i)},\mathbf{G}} \left(\left(S \sum_{i=1}^{S} \mathrm{tr}\Big(\big(\mathbf{X}^{(i)}\mathbf{R}^{(i)}\big)^{\top} \mathbf{X}^{(i)}\mathbf{R}^{(i)} \Big) \right) - 2\Big(S^2 \mathrm{tr}\big(\mathbf{G}^{\top}\mathbf{G}\big) \Big) + \Big(S^2 \mathrm{tr}\big(\mathbf{G}^{\top}\mathbf{G}\big) \Big) \right)$$

$$= \min_{\mathbf{R}^{(i)},\mathbf{G}} S \sum_{i=1}^{S} \left(\mathrm{tr}\Big(\big(\mathbf{X}^{(i)}\mathbf{R}^{(i)}\big)^{\top} \mathbf{X}^{(i)}\mathbf{R}^{(i)} \Big) + \mathrm{tr}\big(\mathbf{G}^{\top}\mathbf{G}\big) - 2\mathrm{tr}\Big(\big(\mathbf{X}^{(i)}\mathbf{R}^{(i)}\big)^{\top} \mathbf{G} \Big) \right)$$

$$\equiv \min_{\mathbf{R}^{(i)},\mathbf{G}} \sum_{i=1}^{S} \left\| \mathbf{G} - \mathbf{X}^{(i)}\mathbf{R}^{(i)} \right\|_F^2$$

□

Classical HA approach can be generalized to the real-world problems as follows:

$$\min_{\mathbf{R}^{(i)},\mathbf{G}} \sum_{i=1}^{S} \left\| f\big(\mathbf{X}^{(i)}\big)\mathbf{R}^{(i)} - \mathbf{G} \right\|_F^2, \quad \text{s.t.} \quad \Big(f\big(\mathbf{X}^{(\ell)}\big)\mathbf{R}^{(\ell)} \Big)^{\top} f\big(\mathbf{X}^{(\ell)}\big)\mathbf{R}^{(\ell)} = \mathbf{I}, \ell = 1{:}S. \tag{2.8}$$

Here, if $f\big(\mathbf{x}\big) = \mathbf{x}$, then we recover the original HA. Further, if $f\big(\mathbf{x}\big)$ denotes any classical fixed kernel function (e.g. Gaussian), then (2.8) is equivalent to Kernel Hyperalignment (KHA) [4, 5]. In addition, if $f\big(\mathbf{x}\big)$ illustrates an SVD-based feature selection, then (2.8) is called SVD Hyperalignment (SVDHA) [12]. As another alternative, (2.8) can be optimized by using different approaches. For instance, Regularized Hyperalignment (RHA) used the Generalized CCA [11], Shared Response Model (SRM) employed Probabilistic CCA [9].

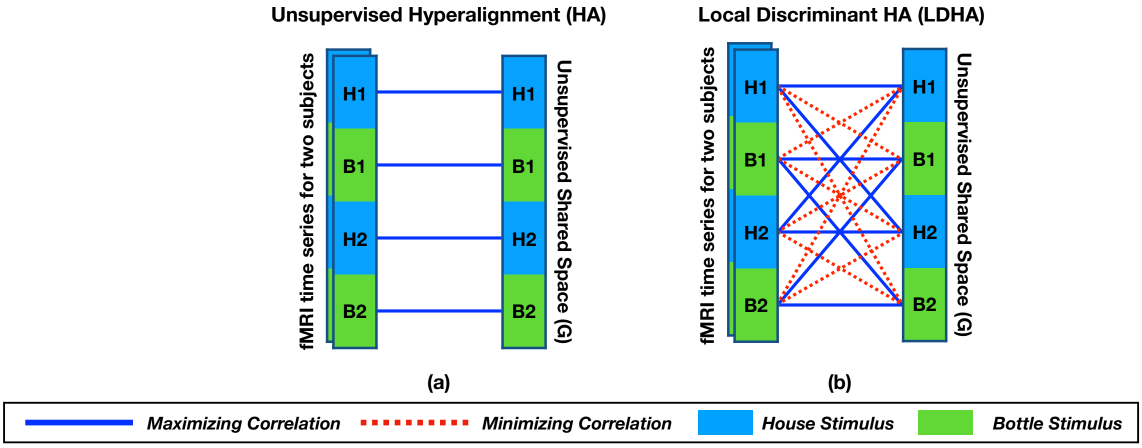

Unsupervised Hyperalignment (HA)
Local Discriminant HA (LDHA)
fMRI time series for two subjects
H1
B1
H2
B2
Unsupervised Shared Space (G)
(a)
(b)
Maximizing Correlation
Minimizing Correlation
House Stimulus
Bottle Stimulus

Figure 2.1 Comparison of unsupervised and supervised HA algorithms

2.3 Local Discriminant Hyperalignment

Since the unsupervised CCA techniques are employed for solving HA problems, the solution may not be optimized for supervised MVP analysis. In other words, CCA just finds a set of mappings to maximize the correlation between the same time-points of functional activities *(in voxel-level)* for all subjects, while it must maximize the correlation between homogeneous stimuli (from the same category) *and also remove the correlation between different categories of stimuli*. Indeed, this is a common problem in Machine Learning. For instance, Linear Discriminant Analysis (LDA) is mostly used rather than Principal Component Analysis (PCA) in the classification analysis, where LDA uses the supervision information such as class labels or similarity between samples for improving the performance of classification methods.

In this section, we introduce Local Discriminant Hyperalignment (LDHA) method, which incorporates the idea of Local Discriminate Analysis (LDA) into CCA [40] in order to improve the performance of the hyperalignment solution. As Figure 2.1 depicted, the idea of locality is defined based on the stimuli categories (class labels) in the train-set, where the proposed method firstly generates two sets for each category of stimuli, i.e. the set of nearest homogeneous stimuli as within-class neighborhoods and the set of stimuli from distinct categories as between-class neighborhoods. Then, these two sets are used to provide a better HA solution, where the correlation between the within-class neighborhoods is maximized, and also the correlation among between-class neighborhoods approaches to zero.

As mentioned before, (2.3) may not be optimum for supervised fMRI analysis. In order to improve the performance of functional alignment, Local Discriminant Hyperalignment (LDHA) uses following objective function:

$$\max_{\mathbf{R}^{(i)},\mathbf{R}^{(j)}} \sum_{i=1}^{S} \sum_{j=i+1}^{S} \text{tr}\Big(\big(\mathbf{R}^{(i)}\big)^{\top}\big(\mathbf{\Delta}^{(i,j)} - \frac{\eta}{T^2}\mathbf{\Omega}^{(i,j)}\big)\mathbf{R}^{(j)}\Big) \tag{2.9}$$

where η is the number of within-class elements, and T denotes all time points. Further, the covariance within-class matrix $\mathbf{\Delta}^{(i,j)} = \Big\{\delta_{mn}^{(i,j)}\Big\}$ and the covariance between-class matrix $\mathbf{\Omega}^{(i,j)} = \Big\{\omega_{mn}^{(i,j)}\Big\}$ are denoted as follows:

$$\delta_{mn}^{(i,j)} = \sum_{\ell=1}^{T}\sum_{k=1}^{T} \alpha_{\ell k} x_{\ell m}^{(i)} x_{kn}^{(j)} + \alpha_{\ell k} x_{\ell n}^{(i)} x_{km}^{(j)} \tag{2.10}$$

$$\omega_{mn}^{(i,j)} = \sum_{\ell=1}^{T}\sum_{k=1}^{T} (1-\alpha_{\ell k}) x_{\ell m}^{(i)} x_{kn}^{(j)} + (1-\alpha_{\ell k}) x_{\ell n}^{(i)} x_{km}^{(j)} \tag{2.11}$$

where $\alpha_{\ell k} = 1$ for within-class elements, otherwise it is zero. Moreover, $x_{mn}^{(i)} \in \mathbb{R}$ is the neural activity for the $m-th$ time point and the $n-th$ voxel in the $i-th$ subject.

Lemma 2.4. *Like the classical CCA, LDHA can be solved as a generalized eigenvalue decomposition problem.*

Proof. We firstly define following matrix by using the covariance matrices:

$$\mathbf{\Phi}^{(i,j)} = \mathbf{\Delta}^{(i,j)} - \frac{\eta}{T^2}\mathbf{\Omega}^{(i,j)}. \tag{2.12}$$

Then, we have:

$$\mathbf{P}^{(i,j)} = \big(\widetilde{\mathbf{\Phi}}^{(i)}\big)^{-1/2}\mathbf{\Phi}^{(i,j)}\big(\widetilde{\mathbf{\Phi}}^{(j)}\big)^{-1/2} \tag{2.13}$$

where $\mathbf{P}^{(i,j)} \neq \mathbf{P}^{(j,i)}$, and $\widetilde{\mathbf{\Phi}}^{(\ell)} = (\mathbf{X}^{(\ell)})^{\top}\mathbf{X}^{(\ell)}$. Now, we can apply rank-m SVD decomposition [42] on $\mathbf{P}^{(i,j)}$ as follows:

$$\mathbf{P}^{(i,j)} \overset{SVD}{=} \mathbf{\Omega}^{(i,j)}\mathbf{\Sigma}^{(i,j)}\big(\mathbf{\Psi}^{(i,j)}\big)^{\top}. \tag{2.14}$$

Finally, we can calculate the mapping as follows:

$$\mathbf{R}^{(i)} = \sum_{j=1}^{S} \big(\widetilde{\mathbf{\Phi}}^{(i)}\big)^{-1/2}\mathbf{\Omega}^{(i,j)} \tag{2.15}$$

In addition, the shared space can be calculated by using (2.7). □

Algorithm 2.1 A template for MVP analysis by using LDHA

```
Input: Train Set $\mathbf{X}^{(i)}, i = 1{:}S$, Test Set $\widehat{\mathbf{X}}^{(j)}, j = 1{:}\hat{S}$:
Output: Classification Model $\pi$, Classification Performance ($ACC$, $AUC$):
Method:
01. For each $\mathbf{X}^{(i)}, i = 1{:}S$, calculating $\mathbf{R}^{(i)}$ by using (2.15).
02. Generating the shared space ($\mathbf{G}$) by utilizing (2.7).
03. Training a classifier $\pi$ by using $\mathbf{X}^{(i)}\mathbf{R}^{(i)}, i = 1{:}S$.
04. For each $\widehat{\mathbf{X}}^{(j)}, j = 1{:}\hat{S}$, calculating $\widehat{\mathbf{R}}^{(i)}$ by using $\mathbf{G}$ and (2.6) [11].
05. Evaluating ($ACC$, $AUC$) the trained classifier $\pi$ by using $\widehat{\mathbf{X}}^{(j)}\widehat{\mathbf{R}}^{(i)}$.
```

Algorithm 2.1 demonstrates a general template for MVP analysis based on LDHA method. Here, $\mathbf{X}^{(i)}, i = 1{:}S$ is the training set for S subjects. Further, $\widehat{\mathbf{X}}^{(j)}, j = 1{:}\hat{S}$ denotes the testing set for $\hat{S}$ subjects. As this algorithm depicted, the procedure of generating the HA template ($\mathbf{G}$) in the training stage is changed, while the template is used in the testing stage such as the unsupervised HA methods [4, 11]. Therefore, we **do not** need the class labels in the testing stage. Indeed, the proposed method in comparison with the unsupervised solutions just generates more optimum HA template for aligning functional neural activities, where this template can maximize the correlation between all stimuli in the same category and minimize the correlation between different categories of stimuli.

2.4 Deep Hyperalignment

This section introduces a new nonlinear methods that can improve three issues in HA problems, i.e., nonlinearity, high-dimensionality, and using a large number of subjects. As Figure 2.2 depicted, we propose a novel kernel approach, which is called Deep Hyperalignment (DHA), where it employs deep network, i.e. multiple stacked layers of nonlinear transformation, as the kernel function, which is parametric and uses rank-m SVD and Stochastic Gradient Descent (SGD) for optimization. Consequently, DHA generates low-runtime on large datasets, and the training data is not referenced when DHA computes the functional alignment for a new subject. Further, DHA is not limited by a restricted fixed representational space because the kernel in DHA is a multi-layer neural network, which can separately implement *any nonlinear function* [13, 41, 42] for each subject to transfer the

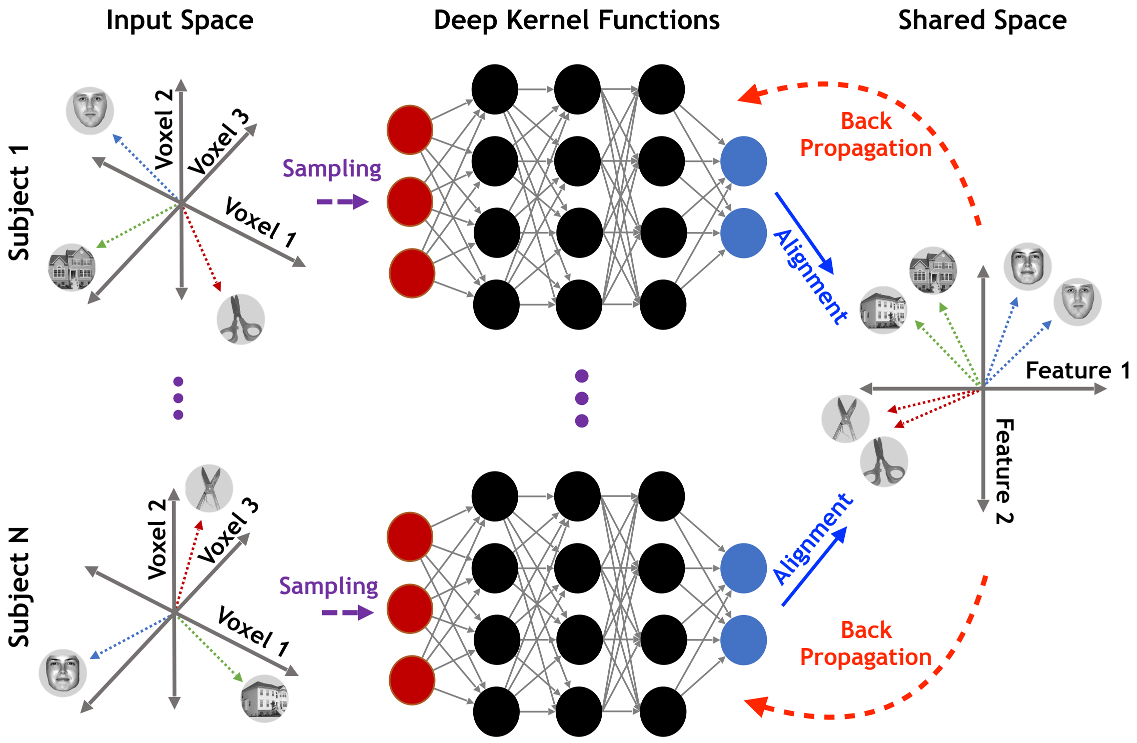


Figure 2.2 Deep Hyperalignment

brain activities to a common space. Based on (2.8), objective function of DHA is defined as follows:

$$\begin{aligned} &\min_{\mathbf{G},\mathbf{R}^{(i)},\boldsymbol{\theta}^{(i)}} \sum_{i=1}^{S} \left\| \mathbf{G} - f_i\big(\mathbf{X}^{(i)};\boldsymbol{\theta}^{(i)}\big)\mathbf{R}^{(i)} \right\|_F^2 \\ &\mathbf{G} = \frac{1}{S}\sum_{j=1}^{S} f_j\big(\mathbf{X}^{(j)};\boldsymbol{\theta}^{(j)}\big)\mathbf{R}^{(j)}, \\ \text{s.t.} \quad &\Big(\mathbf{R}^{(\ell)}\Big)^\top \bigg(\Big(f_\ell\big(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)}\big)\Big)^\top f_\ell\big(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)}\big) + \epsilon\mathbf{I}\bigg)\mathbf{R}^{(\ell)} = \mathbf{I}, \quad \ell = 1{:}S, \end{aligned} \tag{2.16}$$

where $\boldsymbol{\theta}^{(\ell)}=\{\mathbf{W}_m^{(\ell)}, \mathbf{b}_m^{(\ell)}, m=2{:}C\}$ denotes *all parameters in ℓ-th deep network belonged to ℓ-th subject*, $\mathbf{R}^{(\ell)} \in \mathbb{R}^{V_{new}\times V_{new}}$ is the DHA solution for ℓ-*th* subject, $V_{new} \leq V$ denotes the number of features after transformation, the regularized parameter ϵ is a small constant, e.g., 10^{-8}, and deep multi-layer kernel function $f_\ell\big(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)}\big) \in \mathbb{R}^{T\times V_{new}}$ is denoted as follows:

$$f_\ell\big(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)}\big) = \text{mat}\Big(\mathbf{h}_C^{(\ell)}, T, V_{new}\Big), \tag{2.17}$$

where T denotes the number of time points, $C \geq 3$ is number of deep network layers, mat$(\mathbf{x}, m, n)$:$\mathbb{R}^{mn} \to \mathbb{R}^{m\times n}$ denotes the reshape (matricization) function, and $\mathbf{h}_C^{(\ell)} \in \mathbb{R}^{TV_{new}}$ is the output layer of the following multi-layer deep network:

$$\mathbf{h}_m^{(\ell)} = g\Big(\mathbf{W}_m^{(\ell)}\mathbf{h}_{m-1}^{(\ell)} + \mathbf{b}_m^{(\ell)}\Big), \quad \text{where} \quad \mathbf{h}_1^{(\ell)} = \text{vec}\big(\mathbf{X}^{(\ell)}\big) \quad \text{and} \quad m = 2{:}C. \tag{2.18}$$

Here, $g{:}\mathbb{R} \to \mathbb{R}$ is a nonlinear function applied componentwise, $vec{:}\mathbb{R}^{m\times n} \to \mathbb{R}^{mn}$ denotes the vectorization function, consequently $\mathbf{h}_1^{(\ell)} = \text{vec}\big(\mathbf{X}^{(\ell)}\big) \in \mathbb{R}^{TV}$. Notably, this thesis considers both $vec()$ and $mat()$ functions are linear transformations, where $\mathbf{X} \in \mathbb{R}^{m\times n} = \text{mat}\big(\text{vec}(\mathbf{X}), m, n\big)$ for any matrix $\mathbf{X}$. By considering $U^{(m)}$ units in the m-th intermediate layer, parameters of distinctive layers of $f_\ell\big(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)}\big)$ are defined by following properties: $\mathbf{W}_C^{(\ell)} \in \mathbb{R}^{TV_{new}\times U^{(C\text{-}1)}}$ and $\mathbf{b}_C^{(\ell)} \in \mathbb{R}^{TV_{new}}$ for the output layer, $\mathbf{W}_2^{(\ell)} \in \mathbb{R}^{U^{(2)}\times TV}$ and $\mathbf{b}_2^{(\ell)} \in \mathbb{R}^{U^{(2)}}$ for the first intermediate layer, and $\mathbf{W}_m^{(\ell)} \in \mathbb{R}^{U^{(m)}\times U^{(m\text{-}1)}}$, $\mathbf{b}_m^{(\ell)} \in \mathbb{R}^{U^{(m)}}$ and $\mathbf{h}_m^{(\ell)} \in \mathbb{R}^{U^{(m)}}$ for m-th intermediate layer ($3 \leq m \leq C - 1$).

Remark 1. Same as previous approaches for HA problems [4, 9, 11, 12], a DHA solution is not unique. If a DHA template **G** is calculated for a specific HA problem, then **QG** is another solution for that specific HA problem, where $\mathbf{Q} \in \mathbb{R}^{V_{new}\times V_{new}}$ can be any orthogonal matrix. Consequently, if two independent templates $\mathbf{G}_1$, $\mathbf{G}_2$ are trained for a specific dataset, the solutions can be mapped to each other by calculating $\big|\big|\mathbf{G}_2 - \mathbf{Q}\mathbf{G}_1\big|\big|$, where **Q** can be used as a coefficient for functional alignment in the first solution in order to compare its results to the second one. Indeed, $\mathbf{G}_1$ and $\mathbf{G}_2$ are located in different positions on the same contour line [2, 9, 13].

We propose an effective approach for optimizing the DHA objective function by using rank-m SVD and SGD. This method seeks an optimum solution for the DHA objective function (2.16) by using two different steps, which iteratively work in unison. By considering fixed network parameters ($\boldsymbol{\theta}^{(\ell)}$), a mini-batch of neural activities is firstly aligned through the deep network. Then, back-propagation algorithm is used to update the network parameters. The main challenge for solving the DHA objective function is that we cannot seek a natural extension of the correlation object to more than two random variables. Consequently, functional alignments are stacked in a $S \times S$ matrix and maximize a certain matrix norm for that matrix [42].

As the first step, we consider network parameters are in an optimum state. Therefore, the mappings ($\mathbf{R}^{(\ell)}$, $\ell = 1{:}S$) and template (**G**) must be calculated to solve the DHA problem. In order to scale DHA approach, this thesis employs the rank-m SVD [42] of the mapped neural activities as follows:

$$f_\ell\big(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)}\big) \overset{SVD}{=} \boldsymbol{\Omega}^{(\ell)}\boldsymbol{\Sigma}^{(\ell)}\big(\boldsymbol{\Psi}^{(\ell)}\big)^\top, \qquad \ell = 1{:}S \tag{2.19}$$

where $\mathbf{\Sigma}^{(\ell)} \in \mathbb{R}^{m\times m}$ denotes the diagonal matrix with m-largest singular values of the mapped feature $f_\ell\big(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)}\big)$, $\mathbf{\Omega}^{(\ell)} \in \mathbb{R}^{T\times m}$ and $\mathbf{\Psi}^{(\ell)} \in \mathbb{R}^{m\times V_{new}}$ are respectively the corresponding left and right singular vectors. Based on (2.19), the projection matrix for ℓ-th subject can be generated as follows: [42]

$$\begin{aligned}\mathbf{P}^{(\ell)} &= f_\ell\big(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)}\big)\Big(\Big(f_\ell\big(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)}\big)\Big)^\top f_\ell\big(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)}\big)+\epsilon\mathbf{I}\Big)^{-1}\Big(f_\ell\big(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)}\big)\Big)^\top \\ &= \mathbf{\Omega}^{(\ell)}\big(\mathbf{\Sigma}^{(\ell)}\big)^\top\Big(\mathbf{\Sigma}^{(\ell)}\big(\mathbf{\Sigma}^{(\ell)}\big)^\top+\epsilon\mathbf{I}\Big)^{-1}\mathbf{\Sigma}^{(\ell)}\big(\mathbf{\Omega}^{(\ell)}\big)^\top = \mathbf{\Omega}^{(\ell)}\mathbf{D}^{(\ell)}\Big(\mathbf{\Omega}^{(\ell)}\mathbf{D}^{(\ell)}\Big)^\top,\end{aligned} \tag{2.20}$$

where $\mathbf{P}^{(\ell)} \in \mathbb{R}^{T\times T}$ is symmetric and idempotent [42], and diagonal matrix $\mathbf{D}^{(\ell)} \in \mathbb{R}^{m\times m}$ is

$$\mathbf{D}^{(\ell)}\big(\mathbf{D}^{(\ell)}\big)^\top = \big(\mathbf{\Sigma}^{(\ell)}\big)^\top\Big(\mathbf{\Sigma}^{(\ell)}\big(\mathbf{\Sigma}^{(\ell)}\big)^\top+\epsilon\mathbf{I}\Big)^{-1}\mathbf{\Sigma}^{(\ell)}. \tag{2.21}$$

Further, the sum of projection matrices can be defined as follows, where $\widetilde{\mathbf{A}}\widetilde{\mathbf{A}}^\top$ is the Cholesky decomposition [42] of $\mathbf{A}$:

$$\mathbf{A} = \sum_{i=1}^{S}\mathbf{P}^{(i)} = \widetilde{\mathbf{A}}\widetilde{\mathbf{A}}^\top, \quad \text{where} \quad \widetilde{\mathbf{A}} \in \mathbb{R}^{T\times mS} = \big[\mathbf{\Omega}^{(1)}\mathbf{D}^{(1)}\ldots\mathbf{\Omega}^{(S)}\mathbf{D}^{(S)}\big]. \tag{2.22}$$

Lemma 2.5. *By considering* (2.22) *and following mapping function:*

$$\mathbf{R}^{(\ell)} = \Big(\Big(f_\ell\big(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)}\big)\Big)^\top f_\ell\big(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)}\big)+\epsilon\mathbf{I}\Big)^{-1}\Big(f_\ell\big(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)}\big)\Big)^\top\mathbf{G}, \tag{2.23}$$

the objective function of DHA (2.16) *can be rewritten as follows:*

$$\min_{\mathbf{G},\mathbf{R}^{(i)},\boldsymbol{\theta}^{(i)}}\sum_{i=1}^{S}\Big\|\mathbf{G}-f_i\big(\mathbf{X}^{(i)};\boldsymbol{\theta}^{(i)}\big)\mathbf{R}^{(i)}\Big\| \equiv \max_{\mathbf{G}}\Big(\mathrm{tr}\big(\mathbf{G}^\top\mathbf{A}\mathbf{G}\big)\Big). \tag{2.24}$$

Proof. For simplicity, we consider that $f_\ell\big(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)}\big) = \mathbf{Y}^{(\ell)}$. By replacing (2.23), we have:

$$\min_{\mathbf{G},\boldsymbol{\theta}^{(i)},\mathbf{R}^{(i)}}\sum_{i=1}^{S}\Big\|\mathbf{G}-\mathbf{Y}^{(i)}\Big(\big(\mathbf{Y}^{(i)}\big)^\top\mathbf{Y}^{(i)}+\epsilon\mathbf{I}\Big)^{-1}\big(\mathbf{Y}^{(i)}\big)^\top\mathbf{G}\Big\|_F^2$$

Based on (2.20), we have:

$$\begin{aligned}\min_{\mathbf{G}}\sum_{i=1}^{S}\|\mathbf{G}-\mathbf{P}^{(i)}\mathbf{G}\|_F^2 &= \min_{\mathbf{G}}\sum_{i=1}^{S}\|\big(\mathbf{I}-\mathbf{P}^{(i)}\big)\mathbf{G}\|_F^2 = \min_{\mathbf{G}}\sum_{i=1}^{S}\mathrm{tr}\Big(\Big(\big(\mathbf{I}-\mathbf{P}^{(i)}\big)\mathbf{G}\Big)^\top\big(\mathbf{I}-\mathbf{P}^{(i)}\big)\mathbf{G}\Big) \\ &= \min_{\mathbf{G}}\sum_{i=1}^{S}\mathrm{tr}\Big(\mathbf{G}^\top\big(\mathbf{I}-\mathbf{P}^{(i)}\big)^\top\big(\mathbf{I}-\mathbf{P}^{(i)}\big)\mathbf{G}\Big) = \min_{\mathbf{G}}\sum_{i=1}^{S}\mathrm{tr}\Big(\mathbf{G}^\top\big(\mathbf{I}-\mathbf{P}^{(i)}\big)^2\mathbf{G}\Big)\end{aligned}$$

Since $\mathbf{P}^{(i)}$ is idempotent ($\big(\mathbf{P}^{(i)}\big)^2 = \mathbf{P}^{(i)}$) [42, 54], we have:

$$\begin{aligned}
\min_{\mathbf{G}} \sum_{i=1}^{S} \text{tr}\Big(\mathbf{G}^\top \big(\mathbf{I} - \mathbf{P}^{(i)}\big)^2 \mathbf{G}\Big) &= \min_{\mathbf{G}} \sum_{i=1}^{S} \text{tr}\Big(\mathbf{G}^\top \Big(\mathbf{I}^2 + \big(\mathbf{P}^{(i)}\big)^2 - 2\mathbf{I}\mathbf{P}^{(i)}\Big)\mathbf{G}\Big) \\
= \min_{\mathbf{G}} \sum_{i=1}^{S} \text{tr}\Big(\mathbf{G}^\top \Big(\mathbf{I}^2 + \mathbf{P}^{(i)} - 2\mathbf{P}^{(i)}\Big)\mathbf{G}\Big) &= \min_{\mathbf{G}} \sum_{i=1}^{S} \text{tr}\Big(\mathbf{G}^\top \big(\mathbf{I} - \mathbf{P}^{(i)}\big)\mathbf{G}\Big) \\
= \min_{\mathbf{G}} \sum_{i=1}^{S} \text{tr}\Big(\mathbf{G}^\top \mathbf{I}\mathbf{G} - \mathbf{G}^\top \mathbf{P}^{(i)}\mathbf{G}\Big) &= \min_{\mathbf{G}} \sum_{i=1}^{S} \Big(\text{tr}\big(\mathbf{I}\big) - \text{tr}\big(\mathbf{G}^\top \mathbf{P}^{(i)}\mathbf{G}\big)\Big) \\
&= \min_{\mathbf{G}} \Big(SV - \text{tr}\Big(\mathbf{G}^\top \big(\sum_{i=1}^{S} \mathbf{P}^{(i)}\big)\mathbf{G}\Big)\Big)
\end{aligned}$$

Based on (2.22), we have:

$$= \min_{\mathbf{G}} \Big(SV - \text{tr}\Big(\mathbf{G}^\top \mathbf{A}\mathbf{G}\Big)\Big) \equiv \max_{\mathbf{G}} \Big(\text{tr}\big(\mathbf{G}^\top \mathbf{A}\mathbf{G}\big)\Big)$$

□

Based on Lemma 2.5, the first optimization step of DHA problem can be expressed as eigendecomposition of $\mathbf{A}\mathbf{G} = \mathbf{G}\mathbf{\Lambda}$, where $\mathbf{\Lambda} = \big\{\lambda_1 \dots \lambda_T\big\}$ and $\mathbf{G}$ respectively denote the eigenvalues and eigenvectors of $\mathbf{A}$. Further, the matrix $\mathbf{G}$ that we are interested in finding, can be calculated by the left singular vectors of $\widetilde{\mathbf{A}} = \mathbf{G}\widetilde{\mathbf{\Sigma}}\widetilde{\mathbf{\Psi}}^\top$, where $\mathbf{G}^\top\mathbf{G} = \mathbf{I}$ [42, 54]. This thesis utilizes Incremental SVD [55] for calculating these left singular vectors.

Lemma 2.6. *In order to update network parameters as the second step, the derivative of* $\mathbf{Z} = \sum_{\ell=1}^{T} \lambda_\ell$*, which is the sum of eigenvalues of* $\mathbf{A}$*, over the mapped neural activities of* ℓ*-th subject is defined as follows:*

$$\frac{\partial \mathbf{Z}}{\partial f_\ell\big(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)}\big)} = 2\mathbf{R}^{(\ell)}\mathbf{G}^\top - 2\mathbf{R}^{(\ell)}\big(\mathbf{R}^{(\ell)}\big)^\top \Big(f_\ell\big(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)}\big)\Big)^\top. \tag{2.25}$$

Proof. For simplicity, we define $f_\ell\big(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)}\big) = \mathbf{Y}^{(\ell)}$ and the covariance matrix for ℓ-th subject as follows:

$$\widetilde{\mathbf{\Phi}}^{(\ell)} = \Big(\big(\mathbf{Y}^{(\ell)}\big)^\top \mathbf{Y}^{(\ell)} + \epsilon \mathbf{I}\Big) \tag{2.26}$$

Further, the inverse of the covariance matrix is denoted for ℓ-th subject as follows:

$$\mathbf{\Phi}^{(\ell)} = \big(\widetilde{\mathbf{\Phi}}^{(\ell)}\big)^{-1} = \Big(\big(\mathbf{Y}^{(\ell)}\big)^\top \mathbf{Y}^{(\ell)} + \epsilon \mathbf{I}\Big)^{-1} \tag{2.27}$$

where this matrix is symmetric ($\mathbf{\Phi}^{(\ell)} = \left(\mathbf{\Phi}^{(\ell)}\right)^{\top}$). Based on (2.27), the projection (2.20) can be rewritten as follows:

$$\mathbf{P}^{(\ell)} = \mathbf{Y}^{(\ell)}\mathbf{\Phi}^{(\ell)}\left(\mathbf{Y}^{(\ell)}\right)^{\top} \tag{2.28}$$

Further, the final mappings (2.23) can be reformulated as follows:

$$\mathbf{R}^{(\ell)} = \mathbf{\Phi}^{(\ell)}\left(\mathbf{Y}^{(\ell)}\right)^{\top}\mathbf{G} \tag{2.29}$$

Since $\partial\mathbf{Z}/\partial\mathbf{A} = \mathbf{G}\mathbf{G}^{\top}$ [14], we reformulate the left side of (2.25) based on the chain rule as follows:

$$\frac{\partial \mathbf{Z}}{\partial \mathbf{Y}^{(\ell)}_{\alpha\beta}} = \sum_{\mu,\tau=1}^{V_{new}} \frac{\partial \mathbf{Z}}{\partial \mathbf{A}_{\mu\tau}} \frac{\partial \mathbf{A}_{\mu\tau}}{\partial \mathbf{Y}^{(\ell)}_{\alpha\beta}} = \sum_{\mu,\tau=1}^{V_{new}} \left(\mathbf{G}\mathbf{G}^{\top}\right)_{\tau\mu} \frac{\partial \mathbf{A}_{\mu\tau}}{\partial \mathbf{Y}^{(\ell)}_{\alpha\beta}} \tag{2.30}$$

By considering (2.22), the product rule, and this key point that $\mathbf{P}^{(\ell)}$ is the only projection related to $\mathbf{Y}^{(\ell)}$, we have:

$$\begin{aligned}\frac{\partial \mathbf{A}_{\mu\tau}}{\partial \mathbf{Y}^{(\ell)}_{\alpha\beta}} = \frac{\partial \mathbf{P}^{(\ell)}_{\mu\tau}}{\partial \mathbf{Y}^{(\ell)}_{\alpha\beta}} &= \delta_{\mu\beta} \sum_{i=1}^{V_{new}} \mathbf{Y}^{(\ell)}_{\tau i}\mathbf{\Phi}^{(\ell)}_{\alpha i} + \delta_{\tau\beta} \sum_{j=1}^{V_{new}} \mathbf{Y}^{(\ell)}_{\mu j}\mathbf{\Phi}^{(\ell)}_{j\alpha} + \sum_{i,j=1}^{V_{new}} \mathbf{Y}^{(\ell)}_{\mu j}\mathbf{Y}^{(\ell)}_{\tau i} \frac{\partial \mathbf{\Phi}^{(\ell)}_{ji}}{\partial \mathbf{Y}^{(\ell)}_{\alpha\beta}} \\ &= \delta_{\mu\beta}\left(\mathbf{\Phi}^{(\ell)}\left(\mathbf{Y}^{(\ell)}\right)^{\top}\right)_{\alpha\tau} + \delta_{\tau\beta}\left(\mathbf{\Phi}^{(\ell)}\left(\mathbf{Y}^{(\ell)}\right)^{\top}\right)_{\alpha\mu} + \sum_{i,j=1}^{V_{new}} \mathbf{Y}^{(\ell)}_{\mu j}\mathbf{Y}^{(\ell)}_{\tau i} \frac{\partial \mathbf{\Phi}^{(\ell)}_{ji}}{\partial \mathbf{Y}^{(\ell)}_{\alpha\beta}}\end{aligned} \tag{2.31}$$

The last term also can be calculated by using the chain rule as follows:

$$\begin{aligned}\frac{\partial \mathbf{\Phi}^{(\ell)}_{ji}}{\partial \mathbf{Y}^{(\ell)}_{\alpha\beta}} &= \sum_{m,n=1}^{T} \frac{\partial \mathbf{\Phi}^{(\ell)}_{ji}}{\widetilde{\mathbf{\Phi}}^{(\ell)}_{mn}} \frac{\widetilde{\mathbf{\Phi}}^{(\ell)}_{mn}}{\partial \mathbf{Y}^{(\ell)}_{\alpha\beta}} = -\sum_{m,n=1}^{T} \left(\mathbf{\Phi}^{(\ell)}_{jm}\mathbf{\Phi}^{(\ell)}_{ni}\left(\delta_{\alpha m}\mathbf{Y}^{(\ell)}_{\beta n} + \delta_{\alpha n}\mathbf{Y}^{(\ell)}_{\beta m}\right)\right) \\ = -\sum_{n=1}^{T} \mathbf{\Phi}^{(\ell)}_{j\alpha}\mathbf{\Phi}^{(\ell)}_{ni}\mathbf{Y}^{(\ell)}_{\beta n} &- \sum_{m=1}^{T} \mathbf{\Phi}^{(\ell)}_{jm}\mathbf{\Phi}^{(\ell)}_{\alpha i}\mathbf{Y}^{(\ell)}_{\beta m} = -\mathbf{\Phi}^{(\ell)}_{j\alpha}\left(\mathbf{\Phi}^{(\ell)}\left(\mathbf{Y}^{(\ell)}\right)^{\top}\right)_{i\beta} - \mathbf{\Phi}^{(\ell)}_{\alpha i}\left(\mathbf{\Phi}^{(\ell)}\left(\mathbf{Y}^{(\ell)}\right)^{\top}\right)_{j\beta}\end{aligned} \tag{2.32}$$

By applying (2.32) to (2.31), we have:

$$\begin{aligned}\frac{\partial \mathbf{P}^{(\ell)}_{\mu\tau}}{\partial \mathbf{Y}^{(\ell)}_{\alpha\beta}} &= \delta_{\mu\beta}\left(\mathbf{\Phi}^{(\ell)}\left(\mathbf{Y}^{(\ell)}\right)^{\top}\right)_{\alpha\tau} + \delta_{\tau\beta}\left(\mathbf{\Phi}^{(\ell)}\left(\mathbf{Y}^{(\ell)}\right)^{\top}\right)_{\alpha\mu} - \left(\mathbf{\Phi}^{(\ell)}\left(\mathbf{Y}^{(\ell)}\right)^{\top}\right)_{\alpha\mu}\left(\mathbf{Y}^{(\ell)}\mathbf{\Phi}^{(\ell)}\left(\mathbf{Y}^{(\ell)}\right)^{\top}\right)_{\beta\tau} \\ &-\left(\mathbf{\Phi}^{(\ell)}\left(\mathbf{Y}^{(\ell)}\right)^{\top}\right)_{\alpha\tau}\left(\mathbf{Y}^{(\ell)}\mathbf{\Phi}^{(\ell)}\left(\mathbf{Y}^{(\ell)}\right)^{\top}\right)_{\beta\mu} = \delta_{\mu\beta}\left(\mathbf{\Phi}^{(\ell)}\left(\mathbf{Y}^{(\ell)}\right)^{\top}\right)_{\alpha\tau} + \delta_{\tau\beta}\left(\mathbf{\Phi}^{(\ell)}\left(\mathbf{Y}^{(\ell)}\right)^{\top}\right)_{\alpha\mu} - \\ &\left(\mathbf{\Phi}^{(\ell)}\left(\mathbf{Y}^{(\ell)}\right)^{\top}\right)_{\alpha\mu}\left(\mathbf{P}^{(\ell)}\right)_{\beta\tau} - \left(\mathbf{\Phi}^{(\ell)}\left(\mathbf{Y}^{(\ell)}\right)^{\top}\right)_{\alpha\tau}\left(\mathbf{P}^{(\ell)}\right)_{\beta\mu} = \\ &\left(\mathbf{I}-\mathbf{P}^{(\ell)}\right)_{\beta\tau}\left(\mathbf{\Phi}^{(\ell)}\left(\mathbf{Y}^{(\ell)}\right)^{\top}\right)_{\alpha\mu} + \left(\mathbf{I}-\mathbf{P}^{(\ell)}\right)_{\beta\mu}\left(\mathbf{\Phi}^{(\ell)}\left(\mathbf{Y}^{(\ell)}\right)^{\top}\right)_{\alpha\tau}\end{aligned} \tag{2.33}$$

Finally, we can calculate (2.25) as follows:

$$\frac{\partial \mathbf{Z}}{\partial \mathbf{Y}^{(\ell)}_{\alpha\beta}} = \sum_{\mu,\tau=1}^{T} \left(\left(\mathbf{G}\mathbf{G}^\top\right)_{\mu\tau} \left(\mathbf{I} - \mathbf{P}^{(\ell)}\right)_{\mu\beta} \left(\mathbf{\Phi}^{(\ell)} \left(\mathbf{Y}^{(\ell)}\right)^\top\right)_{\alpha\tau} \right)$$
$$+ \sum_{\mu,\tau=1}^{T} \left(\left(\mathbf{G}\mathbf{G}^\top\right)_{\mu\tau} \left(\mathbf{I} - \mathbf{P}^{(\ell)}\right)_{\tau\beta} \left(\mathbf{\Phi}^{(\ell)} \left(\mathbf{Y}^{(\ell)}\right)^\top\right)_{\alpha\mu} \right) = 2\left(\mathbf{\Phi}^{(\ell)} \left(\mathbf{Y}^{(\ell)}\right)^\top \mathbf{G}\mathbf{G}^\top \left(\mathbf{I} - \mathbf{P}^{(\ell)}\right)\right)_{\alpha\beta} \tag{2.34}$$

Consequently, we have:

$$\frac{\partial \mathbf{Z}}{\partial \mathbf{Y}^{(\ell)}} = 2\mathbf{\Phi}^{(\ell)} \left(\mathbf{Y}^{(\ell)}\right)^\top \mathbf{G}\mathbf{G}^\top \left(\mathbf{I} - \mathbf{P}^{(\ell)}\right) \tag{2.35}$$

By considering (2.29), we have:

$$\frac{\partial \mathbf{Z}}{\partial \mathbf{Y}^{(\ell)}} = 2\mathbf{R}^{(\ell)}\mathbf{G}^\top \left(\mathbf{I} - \mathbf{P}^{(\ell)}\right) = 2\mathbf{R}^{(\ell)}\mathbf{G}^\top - 2\mathbf{R}^{(\ell)}\mathbf{G}^\top\mathbf{P}^{(\ell)} \tag{2.36}$$

By applying (2.28), we have:

$$= 2\mathbf{R}^{(\ell)}\mathbf{G}^\top - 2\mathbf{R}^{(\ell)}\mathbf{G}^\top\mathbf{Y}^{(\ell)}\mathbf{\Phi}^{(\ell)}\left(\mathbf{Y}^{(\ell)}\right)^\top \tag{2.37}$$

Since $\mathbf{\Phi}^{(\ell)}$ is symmetric:

$$= 2\mathbf{R}^{(\ell)}\mathbf{G}^\top - 2\mathbf{R}^{(\ell)}\mathbf{G}^\top\mathbf{Y}^{(\ell)}\left(\mathbf{\Phi}^{(\ell)}\right)^\top\left(\mathbf{Y}^{(\ell)}\right)^\top \tag{2.38}$$

and again using (2.29), we finally have:

$$\begin{aligned} \frac{\partial \mathbf{Z}}{\partial \mathbf{Y}^{(\ell)}} &= 2\mathbf{R}^{(\ell)}\mathbf{G}^\top - 2\mathbf{R}^{(\ell)}\left(\mathbf{R}^{(\ell)}\right)^\top\left(\mathbf{Y}^{(\ell)}\right)^\top \\ \implies \frac{\partial \mathbf{Z}}{\partial f_\ell\left(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)}\right)} &= 2\mathbf{R}^{(\ell)}\mathbf{G}^\top - 2\mathbf{R}^{(\ell)}\left(\mathbf{R}^{(\ell)}\right)^\top\left(f_\ell\left(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)}\right)\right)^\top \end{aligned} \tag{2.39}$$

□

Algorithm 2.2 illustrates the DHA method for both training and testing phases. As depicted in this algorithm, (2.23) is just needed as the first step in the testing phase because the DHA template **G** is calculated for this phase based on the training samples. As the second step in the DHA method, the networks' parameters ($\boldsymbol{\theta}^{(\ell)}$) must be updated. This thesis employs the back-propagation algorithm ($backprop()$ function) [56] as well as Lemma 2.6 for this step. In addition, finishing condition is defined by tackling errors in last three iterations, i.e. the average of the difference between each

Algorithm 2.2 Deep Hyperalignment (DHA)

```
Input: Data $\mathbf{X}^{(i)}$, $i = 1{:}S$, Regularized parameter $\epsilon$, Number of layers $C$, Number of units $U^{(m)}$ for $m = 2{:}C$, HA template $\widehat{\mathbf{G}}$ for testing phase (default $\emptyset$), Learning rate $\eta$ (default $10^{-4}$ [42]).
Output: DHA mappings $\mathbf{R}^{(\ell)}$ and parameters $\boldsymbol{\theta}^{(\ell)}$, HA template $\mathbf{G}$ just from training phase
Method:
01. Initialize iteration counter: $m \leftarrow 1$ and $\boldsymbol{\theta}^{(\ell)} \sim \mathcal{N}(0,1)$ for $\ell = 1{:}S$.
02. Construct $f_\ell\big(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)}\big)$ based on (2.17) and (2.18) by using $\boldsymbol{\theta}^{(\ell)}$, $C$, $U^{(m)}$ for $\ell = 1{:}S$.
03. IF ($\widehat{\mathbf{G}} = \emptyset$) THEN        % The first step of DHA: fixed $\boldsymbol{\theta}^{(\ell)}$ and calculating $\mathbf{G}$ and $\mathbf{R}^{(\ell)}$ ↓
04.     Generate $\widetilde{\mathbf{A}}$ by using (2.20) and (2.22).
05.     Calculate $\mathbf{G}$ by applying Incremental SVD [55] to $\widetilde{\mathbf{A}} = \mathbf{G}\widetilde{\mathbf{\Sigma}}\widetilde{\mathbf{\Psi}}^\top$.
06. ELSE
07.     $\mathbf{G} = \widehat{\mathbf{G}}$.
08. END IF
09. Calculate mappings $\mathbf{R}^{(\ell)}$, $\ell = 1{:}S$ by using (2.23).
10. Estimate error of iteration $\gamma_m = \sum_{i=1}^{S}\sum_{j=i+1}^{S}\Big\|f_i\big(\mathbf{X}^{(i)};\boldsymbol{\theta}^{(i)}\big)\mathbf{R}^{(i)} - f_j\big(\mathbf{X}^{(j)};\boldsymbol{\theta}^{(j)}\big)\mathbf{R}^{(j)}\Big\|_F^2$.
11. IF $\big((m > 3)$ and $(\gamma_m \geq \gamma_{m-1} \geq \gamma_{m-2})\big)$ THEN        % This is the finishing condition.
12.     Return calculated $\mathbf{G}$, $\mathbf{R}^{(\ell)}$, $\boldsymbol{\theta}^{(\ell)}(\ell = 1{:}S)$ related to $(m$-$2)$-$th$ iteration.
13. END IF        % The second step of DHA: fixed $\mathbf{G}$ and $\mathbf{R}^{(\ell)}$ and updating $\boldsymbol{\theta}^{(\ell)}$ ↓
14. $\nabla\boldsymbol{\theta}^{(\ell)} \leftarrow \text{backprop}\Big({}^{\partial\mathbf{Z}}/_{\partial f_\ell(\mathbf{X}^{(\ell)};\boldsymbol{\theta}^{(\ell)})}, \boldsymbol{\theta}^{(\ell)}\Big)$ by using (2.25) for $\ell = 1{:}S$.
15. Update $\boldsymbol{\theta}^{(\ell)} \leftarrow \boldsymbol{\theta}^{(\ell)} - \eta\nabla\boldsymbol{\theta}^{(\ell)}$ for $\ell = 1{:}S$ and then $m \leftarrow m + 1$
16. SAVE all DHA parameters related to this iteration and GO TO Line 02.
```

pair correlations of aligned functional activities across subjects (γ_m for last three iterations). In other words, DHA will be finished if the error rates in the last three iterations are going to be the worst. Further, a structure (nonlinear function for componentwise, and numbers of layers and units) for the deep network can be selected based on the optimum-state error (γ_{opt}) generated by training samples across different structures (see Experiment Schemes in the supplementary materials).

2.5 Experiments

The empirical studies are reported in this section. Like previous studies [4, 9, 11, 12, 16], we employ the linear ν-SVM algorithms [57] for generating the classification model. Indeed, we use the binary ν-SVM for datasets with just two categories of stimuli and multi-class ν-SVM [4, 5, 57] with one-vs-all strategy as the multi-class approach. In addition, leave-one-subject-out cross-validation is utilized for partitioning all datasets except data 107 to the training set and testing set. In dataset 107, we have used leave-four-subject-out cross-validation because there are 48 subjects and the number of samples in one subject is not enough for testing set. Different HA methods are employed for functional aligning, and then the mapped neural activities are used to generate the classification model. The performances of the proposed methods are compared with the ν-SVM algorithm as the baseline, where the features are used after anatomical alignment without applying any hyperalignment mapping. The performance of other classification techniques will be analyzed in Chapter 5. Furthermore, the performances of the Regularized HA (RHA) [11] is reported as the basic hyperalignment technique, where RHA algorithm is optimized by employing novel Generalized CCA approach proposed in [54]. In addition, regularized parameters (α, β) in RHA are optimally assigned based on [11]. Moreover, Linear Discriminant Analysis (LDA) is employed for functional alignment as a supervised alternative to compare with LDHA. Also, the result of Shared Response Model (SRM) [9, 18] is compared with the proposed methods. As another deep-learning-based alternative for functional alignment, the performance of CAE [16] is also compared with the proposed method. Like the original paper [16], this thesis employs $k_1 = k_3 = \{5, 10, 15, 20, 25\}$, $\rho = \{0.1, 0.25, 0.5, 0.75, 0.9\}$, $\lambda = \{0.1, 1, 5, 10\}$. Then, aligned neural activities (by using CAE) are applied to the classification algorithm same as other HA techniques. This thesis follows the CAE setup to set the same settings in the DHA method. Consequently, three hidden layers ($C = 5$) and the regularized parameters $\epsilon = \{10^{-4}, 10^{-6}, 10^{-8}\}$ are employed in the DHA method. In addition, the number of units in the intermediate layers are considered $U^{(m)} = KV_{new}$, where $m = 2{:}C\text{-}1$, C is the number of layers, V_{new} denotes the number of aligned features, and K is the number of stimulus categories in each dataset[1]. Further, three distinctive activation functions are employed, i.e. Sigmoid ($\mathrm{g}(\mathbf{x}) = {}^{1}/_{1+\exp(-\mathbf{x})}$), Hyperbolic ($\mathrm{g}(\mathbf{x}) = \tanh(\mathbf{x})$), and Rectified Linear Unit or ReLU ($\mathrm{g}(\mathbf{x}) = \ln(1 + \exp(\mathbf{x}))$). In this thesis, the optimum parameters for DHA and CAE methods are reported for each dataset. Moreover, all algorithms are implemented

[1] Although we can use any settings for DHA, we empirically figured out this setting is acceptable to seek an optimum solution. Indeed, we followed CAE setup in the network structure but used the number of categories (K) rather than a series of parameters. In the current format of DHA, we just need to set the regularized constant and the nonlinear activation function, while a wide range of parameters must be set in the CAE.

by Python 3 on a PC with certain specifications[1] by authors in order to generate experimental results.

2.5.1 Performance Analysis

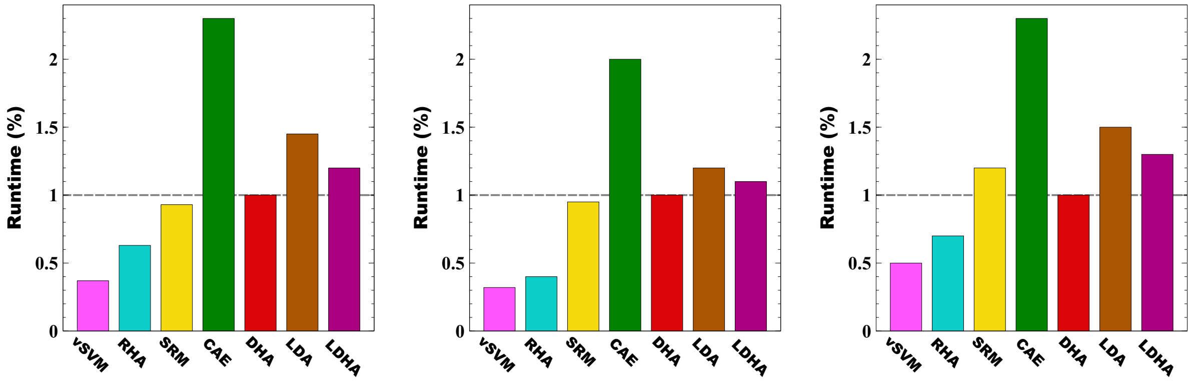
Like previous studies [2, 9, 16, 18], numbers of aligned features are considered equal to $V_{new} = \min\big(V, T\big)$ in this section, where this setting can rapidly improve stability of aligned features [9]. Table 2.1 illustrates the benchmarking of different HA methods based on post-aligned classification accuracy in percentage (%). In data R105 and W105, the best results for CAE are generated by the following parameters $k_1 = k_3 = 25, \rho = 0.9, \lambda = 5$ and for DHA by using $\epsilon = 10^{-6}$ and Sigmoid function. In dataset R107 and W107, the best results for CAE are generated by the following parameters $k_1 = k_3 = 10, \rho = 0.5, \lambda = 10$ and for DHA by using $\epsilon = 10^{-6}$ and ReLU function. In data R232 and W232, the best results for CAE are generated by following parameters $k_1 = k_3 = 20, \rho = 0.9, \lambda = 5$ and for DHA by using $\epsilon = 10^{-8}$ and Sigmoid function. In data W001, the best results for CAE are generated by following parameters $k_1 = k_3 = 10, \rho = 0.75, \lambda = 1$ and for DHA by using $\epsilon = 10^{-4}$ and ReLU function. In data W002D and W002P, the best results for CAE are generated by following parameters $k_1 = k_3 = 15, \rho = 0.5, \lambda = 5$ and for DHA by using $\epsilon = 10^{-4}$ and Sigmoid function. In data W005, the best results for CAE are generated by following parameters $k_1 = k_3 = 20, \rho = 0.75, \lambda = 1$ and for DHA by using $\epsilon = 10^{-8}$ and Hyperbolic function. In the rest of datasets, DHA with Sigmoid function generates better results in comparison with other nonlinear activation functions. In data W011D, the best results for CAE are generated by following parameters $k_1 = k_3 = 5, \rho = 0.1, \lambda = 10$ and for DHA by using $\epsilon = 10^{-4}$. In dataset W011P, we have used the same parameters except $k_1 = k_3 = 10$. In dataset W017, the best results for CAE are generated by following parameters $k_1 = k_3 = 10, \rho = 0.9, \lambda = 5$ and for DHA by using $\epsilon = 10^{-6}$. In data W052R and W052W, the best results for CAE are generated by following parameters $k_1 = k_3 = 25, \rho = 0.2, \lambda = 0.1$ and for DHA by using $\epsilon = 10^{-4}$. In dataset W102, the best results for CAE are generated by the following parameters $k_1 = k_3 = 5, \rho = 0.5, \lambda = 5$ and for DHA by using $\epsilon = 10^{-4}$. In data W116A and W116V, the best results for CAE are generated by the following parameters $k_1 = k_3 = 20, \rho = 0.25, \lambda = 1$ and for DHA by using $\epsilon = 10^{-8}$. In dataset W164, the best results for CAE are generated by the following parameters $k_1 = k_3 = 15, \rho = 0.24, \lambda = 0.1$ and for DHA by using $\epsilon = 10^{-6}$. In data W231, the best results for CAE are generated

[1] CPU = Xeon E5-2630, RAM = 64GB, GPU = GeForce TITAN X, OS = KDE Neon 16.04.3, CUDA = 9.0, CuDNN = 7.0.5, Python = 3.6.5, Pip = 9.0.3, Numpy = 1.14.2, Scipy = 1.0.1, Scikit-Learn = 0.19.1, Tensorflow = 1.7.0, Theano = 0.9.0.

Table 2.1 Accuracy of HA methods in post-alignment classification (max±std)

			Unsupervised			Supervised	
Datasets	ν-SVM	RHA	SRM	CAE	DHA	LDA	LDHA
R105	18.21±2.32	24.90±0.04	35.64±0.05	38.06±0.06	43.76±0.23	37.49±0.03	**52.08±0.11**
R107	27.71±3.86	**82.41±0.63**	40.51±3.32	45.08±0.34	45.87±0.01	38.06±0.32	49.26±0.62
R232	28.12±1.87	31.39±1.20	37.42±0.82	42.75±0.93	45.36±0.07	41.09±0.18	**47.24±0.52**
W001	26.45±0.31	35.74±0.21	35.00±0.64	39.57±0.33	42.14±0.14	32.81±0.94	**45.95±0.97**
W002D	63.81±2.09	66.72±1.12	73.02±0.60	76.22±0.08	**80.04±0.00**	64.95±0.37	77.28±0.53
W002P	65.21±1.51	70.55±0.98	71.41±0.71	79.48±0.06	**82.56±0.01**	68.70±1.04	79.29±0.37
W005	33.59±1.33	42.05±0.24	50.32±0.79	53.24±0.15	**60.32±0.05**	42.00±0.10	59.12±0.62
W011D	42.70±0.90	62.88±1.70	60.60±0.57	65.07±0.13	70.43±0.05	67.83±0.81	**71.99±0.16**
W011W	30.85±0.72	35.61±0.17	37.29±0.78	36.03±0.02	39.05±0.02	34.92±0.79	**41.22±0.28**
W017	20.63±0.94	22.87±0.02	32.83±0.17	**48.96±0.14**	40.26±0.10	43.16±1.03	42.19±1.20
W052R	52.40±1.42	59.93±0.13	63.27±0.39	70.62±1.04	**78.45±0.92**	67.43±1.5	69.69±0.93
W052W	54.07±0.82	60.92±0.06	60.07±0.58	69.92±0.03	**74.40±0.04**	65.02±0.81	70.37±1.08
W102	50.50±0.94	57.56±0.62	69.44±1.31	79.21±0.72	**83.92±0.24**	57.10±0.94	68.02±0.69
W105	16.81±1.77	24.65±0.62	30.06±0.19	35.49±0.26	40.97±0.07	33.92±0.59	**44.66±0.37**
W107	30.69±2.04	47.42±0.94	49.52±0.95	57.04±0.27	69.32±0.01	49.03±0.71	**71.39±0.76**
W116A	59.30±2.72	65.19±0.38	68.00±0.41	75.29±0.50	**85.03±0.05**	62.39±0.93	61.38±0.85
W116V	60.16±0.23	62.27±0.4	65.27±0.83	77.24±0.30	**89.92±0.01**	65.49±0.61	70.35±0.91
W164	53.54±0.53	64.68±0.65	73.82±0.99	70.16±0.18	**91.45±0.03**	69.23±0.22	82.17±0.35
W231	30.62±1.20	58.55±0.54	62.21±0.88	63.42±0.95	67.15±0.05	61.39±0.62	**71.18±0.72**
W232	26.79±0.52	40.21±0.73	47.66±0.29	50.37±0.30	55.17±0.06	35.14±0.84	**67.28±0.93**

by the following parameters $k_1 = k_3 = 15, \rho = 0.75, \lambda = 1$ and for DHA by using $\epsilon = 10^{-4}$. It is worth noting that our empirical studies show that DHA with Sigmoid activation function mostly generates acceptable and stable results in comparison with other activation functions. As Table 2.1 demonstrated, the performances of classification analysis without HA method are significantly low and near to random sampling. Further, while DHA has generated better performance in comparison with other unsupervised methods, LDHA significantly improves the performance of classification analysis, when it employs supervision information to align the multi-subject neural activities. Indeed, DHA illustrates better performance for binary datasets, and LDHA has better accuracy for multi-class problems.



(a) R105 (b) R107 (c) R203

Figure 2.3 Runtime Analysis

2.5.2 Runtime Analysis

This section analyzes the runtime of the proposed approaches and other HA methods by employing ROI-based datasets. As mentioned before, all results in this section are generated by using a PC with certain specifications. Figure 2.3 illustrates the runtime of the mentioned techniques, where runtime of other methods are scaled based on the DHA (runtime of DHA is considered as the unit). As depicted in this figure, CAE generated the worst runtime because it concurrently employs modified versions of SRM and SearchLight for functional alignment. Further, LDA also includes high time complexity because of it must apply matrix decomposition for each category separately. By considering the performance of the DHA method in the previous sections, it generates acceptable runtime among unsupervised approaches. Further, LDHA has better runtime in comparison with LDA because it uses the supervised common space and does not need to apply matrix decomposition for each category separately. It is worth noting that runtime of whole-brain datasets has the same tendency.

2.6 Conclusion

One of the main challenges in fMRI studies is using multi-subject datasets. On the one hand, the multi-subject analysis is necessary to estimate the validity of the generated results across subjects. On the other hand, analyzing multi-subject fMRI data requires accurate functional alignment between neuronal activities of different subjects for improving the performance of the final results. Hyperalignment (HA) is one of the most effective functional alignment methods, which can be formulated as a CCA problem for aligning neural activities of different subjects to a common space.

As the first challenge, the HA solution in MVP analysis may not be optimum because it mostly utilizes the unsupervised CCA techniques for functional alignment. This thesis proposes the Local Discriminant Hyperalignment (LDHA) as a novel supervised HA solution, which employs the concept of locality in machine learning for improving the performances of both functional alignment and MVP analysis. In a nutshell, the proposed method firstly generates two sets for each category of stimuli, i.e. the set of homogeneous stimuli as within-class neighborhoods and the set of stimuli from distinct categories as between-class neighborhoods. Then, these two sets are used to provide a better HA solution, where the correlation between the homogeneous stimuli is maximized, and also the correlation between different categories of stimuli is near to zero.

As the second contribution, we also extended a deep approach for hyperalignment methods in order to provide accurate functional alignment in multi-subject fMRI analysis. Deep Hyperalignment (DHA) can handle fMRI datasets with nonlinearity, high-dimensionality (broad ROI), and a large number of subjects. Indeed, DHA is parametric and uses rank-m SVD and stochastic gradient descent for optimization. Therefore, DHA generates low-runtime on large datasets, and DHA does not require the training data when the functional alignment is computed for a new subject. Further, DHA is not limited by a restricted fixed representational space because the kernel in DHA is a multi-layer neural network, which can separately implement any nonlinear function for each subject to transfer the brain activities to a common space.

Experimental studies on multi-subject fMRI analysis confirm that while DHA method achieves superior performance to other unsupervised approaches, LDHA can significantly improve the performance of analysis, when supervised information is available.

Chapter 3. Snapshots Analysis and Multi-Region Ensemble Learning

The fMRI techniques visualize the neural activities by measuring the level of oxygenation or deoxygenation in the human brain, which is called Blood Oxygen Level Dependent (BOLD) signals. Technically, these signals can be represented as time series for each subject. Most of the MVP techniques directly analyze these noisy and sparse time series for understanding which patterns are demonstrated for different stimuli.

The main idea of the proposed method is so simple. Instead of analyzing whole of the time series, the proposed approach estimates a snapshot of brain image for each stimulus when the level of using oxygen is maximized. As a result, this method can automatically decrease the sparsity of brain image. The proposed method is applied in three stages: firstly, snapshots of brain image are selected by finding local maximums in the smoothed version of the design matrix. Then, features are generated in three steps, including normalizing to standard space, segmenting the snapshots in the form of automatically detected anatomical regions, and removing noise by Gaussian smoothing in the level of ROIs. Finally, we propose two learning approaches. Indeed, extracted features can be analyzed by using both unsupervised learning and supervised learning. This thesis proposed a cluster ensemble approach in order to apply unsupervised learning, where similarities or distances between neural activities can be compared across subjects. As the supervised alternative, we apply an ensemble classification (i.e., bagging technique) on binary $\ell1$-regularized SVM classifiers, where they are created by employing each of neural activities in the level of anatomical regions, i.e., each snapshot represents neural activities for a unique stimulus.

3.1 Unsupervised Learning

As an unsupervised method, Clustering discovers meaningful patterns in the non-labeled data sets. There is a wide range of studies, which try to increase the performance of clustering algorithms. For instance, Zhang et al. introduced a multi-manifold regularized nonnegative matrix factorization framework (MMNMF) which can preserve the locally geometrical structure of the manifolds for multi-view clustering [58]. Anyway, individual clustering algorithms provide different accuracies in a complex data set because they generate the clustering results by optimizing a local or global function instead of natural relations between data points in each data set [59, 60].

Generally, a cluster ensemble has two important steps: Firstly, generating individual clustering results by using different algorithms and changing the number of their partitions. Then, combining the primary results and generating the final ensemble. This step is performed by consensus functions (aggregating mechanism) [59, 61].

The idea that not all partitions are suitable for cooperating to generate the final clustering was proposed in CES [62]. Instead of combing all achieved individual results, CES can combine a selected group of best individual results according to consensus metric(s) from the ensemble committee in order to improve the accuracy of final results [62–65]. Fern and Lin developed a method to effectively select individual clustering results for ensemble and the final decision [62]. Azimi et al. proved that diversity maximization is not an effective approach in some real-world applications. They explored that the thresholding procedure must be done based on the complexity and quality of data sets [65]. Jia et al. proposed SIM for diversity measurement, which works based on the Normalized Mutual Information (NMI) [66]. Romano et al. proposed Standardized Mutual Information (SMI) for evaluating clustering results [67].

Yousefnezhad et al. introduced independency metric instead of quality metric for evaluating the process of solving a problem in the CES [36, 38]. Alizadeh et al. have concluded the disadvantages of NMI as a symmetric criterion. They used the APMM[1] and Maximum (MAX) metrics to measure diversity and stability, respectively, and suggested a new method for building a co-association matrix from a subset of base cluster results [63, 64]. Alizadeh et al. introduced Wisdom of Crowds Cluster Ensemble (WOCCE), which is a novel method base on a theory in social science [64]. Although, this method can generate high performance and more stable results in comparison with other CES methods, using a wide range of thresholds and employing different types of clustering algorithms for generating individual results are two main problems in this method. Alizadeh et al. used A3, which is based on Shannon's entropy, for diversity evaluation; and Basic Parameter Independency (BPI), which uses initialized values of individual clustering algorithms such as random seeds in the first iterative of k-means, for independency evaluation. In addition, they introduced the feedback mechanism for generating the high-quality results [64].

[1] Alizadeh-Parvin-Moshki-Minaei

There are some cluster ensemble approaches focused on rest-mode fMRI analysis. Baumgartner et al. introduced a resampling technique to validate the results of exploratory fuzzy clustering analysis [68]. Sato et al. introduced a novel approach called cluster Granger analysis (CGA) to study connectivity between ROIs. The main aim of this method was to employ multiple eigen–time series in each ROI to avoid temporal information loss during identification of Granger causality [69]. Bellec et al. developed a generic statistical framework to quantify the stability of such resting-state networks (RSNs), which was implemented with k-means clustering [70]. Galdi et al. evaluated the applicability of clustering based techniques to the problem of feature extraction in resting state fMRI analysis [71].

As a graph based clustering methods, spectral clustering generates high-performance results when it is applied to different applications; i.e. from image segmentation to community detection arena. Kuo et al. introduced a new method for automating the process of Laplacian creation in the medical applications; especially for fMRI segmentation where this method used standard Laplacians perform poorly [72]. Chen et al. proposed a clustering algorithm which is based graph clustering and optimizing an appropriate weighted objective, where larger weights are given to observations (edge or no-edge between a pair of nodes) with lower uncertainty [12]. Gao et al. introduced a graph-based consensus maximization (BGCM) method for combining multiple supervised and unsupervised models. This method consolidated a classification solution by maximizing the consensus among both supervised predictions and unsupervised constraints [73].

3.2 Supervised Learning

There are three different types of studies for decoding stimuli in the human brain. Pioneer studies just focused on recognizing special regions of the human brain, such as inanimate objects [74], faces [75], visually illustration of words [76], body parts [77], and visual objects [78]. Although they proved that different stimuli can provide distinctive responses in the brain regions, they cannot find the deterministic locations (or patterns) related to each category of stimuli.

The next group of studies developed correlation techniques in order to understand the similarity (or difference) between distinctive stimuli. Haxby et al. employed brain patterns located in Fusiform Face Area (FFA) and Parahippocampal Place Area (PPA) in order to analyze correlations between different categories of visual stimuli, i.e. gray-scale images of faces, houses, cats, bottles, scissors, shoes, chairs, and scrambled (nonsense) photos [78]. Kamitani and Tong studied the correlations of low-level visual features in the visual cortex (V1–V4) [79]. In similar studies, Haynes et al. analyzed distinctive mental states [80] and more abstract brain patterns such as intentions [81]. Rice et al. proved that not only the brain responses are different based on the categories of the stimuli but also they are correlated based on different properties of the stimuli. They extracted the properties of visual stimuli (photos of objects) and calculated the correlations between these properties and the brain responses. They separately reported the correlation matrices for different human faces and different objects (houses, chairs, etc.) [82].

The last group of studies proposed the MVPA techniques for predicting the category of visual stimuli. Cox et al. utilized linear and non-linear versions of Support Vector Machine (SVM) algorithm [83]. In order to decode the brain patterns, some studies [21, 22, 84] employed classical feature selection (ranking) techniques, such as Principal Component Analysis (PCA) [84], Linear Discriminant Analysis (LDA) [21], or Independent Component Analysis (ICA) [22], that these method are mostly used for analyzing rest-state fMRI datasets. Recent studies proved that not only these techniques cannot provide stable performance in the task-based fMRI datasets [9, 16] but also they had spatial locality issue, especially when they were used for whole brain functional analysis [16]. Norman et al. argued for using SVM and Gaussian Naive Bayes classifiers [85]. Kay et al. studied decoded orientation, position and object category from the brain activities in visual cortex [86]. Mitchell et al. introduced a new method in order to predict the brain activities associated with the meanings of nouns [87]. Miyawaki et al. utilized a combination of multiscale local image decoders in order to reconstruct the visual images from the brain activities [88]. In order to generalize the testing procedure for task-based fMRI datasets, Kriegeskorte et al. proved that the data in testing must have no role in the procedure of generating an MVPA model [89].

There are also some studies that focused on sparse learning techniques. Yamashita et al. developed Sparse Logistic Regression (SLR) in order to improve the performance of classification models [90]. Carroll et al. employed the Elastic Net for prediction and interpretation of distributed neural activity with sparse models [91]. Varoquaux et al. proposed a small-sample brain mapping by using sparse recovery on spatially correlated designs with randomization and clustering. Their method is applied on small sets of brain patterns for distinguishing different categories based on a one-versus-one strategy [92].

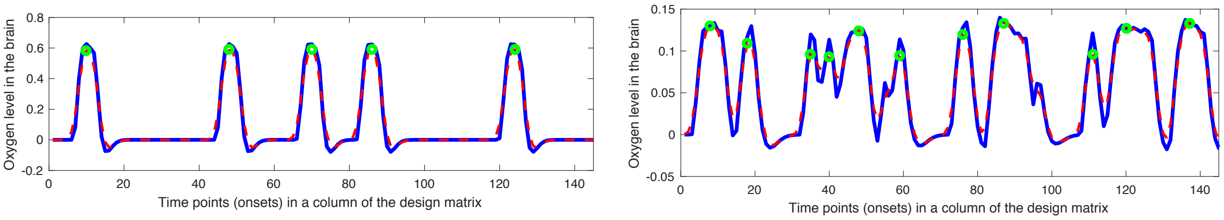
As the first modern approaches for decoding visual stimuli, Anderson and Oates applied nonlinear Artificial Neural Network (ANN) on brain responses [23]. McMenamin et al. studied subsystems underlie Abstract-Category (AC) recognition and priming of objects (e.g., cat, piano) and Specific-Exemplar (SE) recognition and priming of objects (e.g., a calico cat, a different calico cat, a grand piano, etc.). Technically, they applied SVM on manually selected ROIs in the human brain for generating the visual stimuli predictors [25]. Mohr et al. compared four different classification methods, i.e. L1/L2 regularized SVM, the Elastic Net, and the Graph Net, for predicting different responses in the human brain. They show that L1-regularization can improve classification performance while simultaneously providing highly specific and interpretable discriminative activation patterns [24]. Osher et al. proposed a network (graph) based approach by using anatomical regions of the human brain for representing and classifying the different visual stimuli responses (faces, objects, bodies, scenes) [93].

3.3 Snapshots Selection

fMRI time series collected from a subject can be denoted by $\mathbf{F} \in \mathbb{R}^{t \times m}$, where t is the number of time samples, and m denotes the number of voxels. Same as previous studies [23–25, 78], $\mathbf{F}$ can be formulated by a linear model as follows:

$$\mathbf{F} = \mathbf{D}(\widehat{\boldsymbol{\beta}})^{\intercal} + \varepsilon \tag{3.1}$$

where $\mathbf{D} \in \mathbb{R}^{t \times p}$ denotes the design matrix, ε is the noise (error of estimation), $\widehat{\boldsymbol{\beta}} \in \mathbb{R}^{m \times p}$ denotes the sets of correlations (estimated regressors) between voxels. The design matrix can be denoted by $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_i, \dots, \mathbf{d}_p\}$, and the sets of correlations can be defined by $\widehat{\boldsymbol{\beta}} = \{\widehat{\boldsymbol{\beta}}_1, \widehat{\boldsymbol{\beta}}_2, \dots, \widehat{\boldsymbol{\beta}}_i, \dots, \widehat{\boldsymbol{\beta}}_p\}$. Here, $\mathbf{d}_i \in \mathbb{R}^t$ and $\widehat{\boldsymbol{\beta}}_i \in \mathbb{R}^m$ are the column of design matrix and the set of correlations for $i\text{-}th$ category, respectively. p is also the number of all categories in the experiment $\mathbf{F}$. In fact, each category (independent tasks) contains a set of homogeneous visual stimuli. In addition, the nonzero voxels in $\widehat{\boldsymbol{\beta}}_i$ represents the location of all active voxels for the $i\text{-}th$ category [94]. As an example, imagine during a unique session for recognizing visual stimuli, if a subject watches 4 photos of cats and 3 photos of houses, then the design matrix contains two columns; and there are also two sets of correlations between voxels, i.e. one for watching cats and another for watching houses. Indeed, the final goal of this section is extracting 7 snapshots of the brain image for the 7 stimuli in this example.



(a) Block-design experiment

(b) Event-related experiment

Figure 3.1 Examples of smoothed design matrices

The design matrix can be classically calculated by convolution of time samples (or onsets: $\mathbf{S} = \{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_i, \dots, \mathbf{S}_p\}$) and $\boldsymbol{\Xi}$ as the Hemodynamic Response Function (HRF) signal, $\mathbf{d}_i = \mathbf{S}_i * \boldsymbol{\Xi} \implies \mathbf{D} = \mathbf{S} * \mathbf{H}$ [78, 94]. In addition, there is a wide range of solutions for estimating $\widehat{\boldsymbol{\beta}}$ values. This thesis uses the classical method Generalized Least Squares (GLS) [94] for estimating the $\widehat{\boldsymbol{\beta}}$ values where $\boldsymbol{\Sigma}$ is the covariance matrix of the noise ($Var(\varepsilon) = \boldsymbol{\Sigma}\sigma^2 \neq \mathbf{I}\sigma^2$):

$$\widehat{\boldsymbol{\beta}} = \left(\left(\mathbf{D}^{\intercal}\boldsymbol{\Sigma}^{-1}\mathbf{D}\right)^{-1}\mathbf{D}^{\intercal}\boldsymbol{\Sigma}^{-1}\mathbf{F}\right)^{\intercal} \tag{3.2}$$

It is worth noting that we will present a new deep approach for estimating (3.2) in Chapter 4 but we use the classical GLS approach in this section. Here, each local maximum in $\mathbf{d}_i$ represents a location where the level of using oxygen is so high. In other words, the stimulus happens in that location. Since $\mathbf{d}_i$ mostly contains small spikes (especially for event-related experiments), it cannot be directly used for finding these local maximums. Therefore, this thesis employs a Gaussian kernel for smoothing the $\mathbf{d}_i$ signal. Now, the interval $\widehat{\mathbf{G}}$ is defined as follows for generating the kernel:

$$\widehat{\mathbf{G}} = \left\{ \exp\left(\frac{-\widehat{\mathbf{g}}^2}{2\sigma_G^2}\right) \middle| \, \widehat{\mathbf{g}} \in \mathbb{Z} \text{ and } -2\lceil\sigma_G\rceil \leq \widehat{\mathbf{g}} \leq 2\lceil\sigma_G\rceil \right\} \tag{3.3}$$

where $\sigma_G > 0$ denotes a positive real number; $\lceil . \rceil$ is the ceiling function; and $\mathbb{Z}$ denotes the set of integer numbers. Gaussian kernel is also defined by normalizing $\widehat{\mathbf{G}}$ as follows:

$$\mathbf{G} = \frac{\widehat{\mathbf{G}}}{\sum_j \widehat{\mathbf{g}}_j} \tag{3.4}$$

where $\sum_j \widehat{\mathbf{g}}_j$ is the sum of all elements in the interval $\widehat{\mathbf{G}}$. This section defines the smoothed version of the design matrix by applying the convolution of the Gaussian kernel $\mathbf{G}$ and each column of the design matrix ($\mathbf{d}_i$) as follows:

$$\phi_i = \mathbf{d}_i * \mathbf{G} = (\mathbf{S}_i * \boldsymbol{\Xi}) * \mathbf{G} \tag{3.5}$$

$$\boldsymbol{\Phi} = \{\phi_1, \phi_2, \dots, \phi_p\} \tag{3.6}$$

where $\phi_i = f\left(\mathbf{S}_i, \mathbf{\Xi}, \mathbf{G}\right)$. Since the level of smoothness in $\mathbf{\Phi}$ is related to the positive value in (3.3), $\sigma_G = 1$ is heuristically defined to generate the optimum level of smoothness in the design matrix. The general assumption here is the $0 < \sigma_G < 1$ can create design matrix, which is sensitive to small spikes. Further, $\sigma_G > 1$ can rapidly increase the level of smoothness, and remove some weak local maximums, especially in the event-related fMRI datasets. Figure 3.1 illustrates two examples of the smoothed columns in the design matrix. The local maximum points in the ϕ_i can be calculated as follows:

$$\mathbf{S}_i^* = \left\{ \underset{\mathbf{S}_i}{\text{arg}\phi_i} \;\middle|\; \frac{\partial \phi_i}{\partial \mathbf{S}_i} = 0 \quad and \quad \frac{\partial^2 \phi_i}{\partial \mathbf{S}_i \mathbf{S}_i} > 0 \right\} \tag{3.7}$$

where $\mathbf{S}_i^* \subset \mathbf{S}_i$ denotes the set of time points for all local maximums in ϕ_i. The sets of maximum points for all categories can be denoted as follows:

$$\mathbf{S}^* = \{\mathbf{S}_1^*, \mathbf{S}_2^*, \ldots, \mathbf{S}_i^*, \ldots, \mathbf{S}_p^*\} \tag{3.8}$$

As mentioned before, the fMRI time series can be also denoted by $\mathbf{F}^\intercal = \{\mathbf{f}_1^\intercal, \mathbf{f}_2^\intercal, \ldots, \mathbf{f}_j^\intercal, \ldots, \mathbf{f}_t^\intercal\}$, where $\mathbf{f}_j^\intercal \in \mathbb{R}^m$ is all voxels of fMRI dataset in the $j\text{-}th$ time point. Now, the set of snapshots can be formulated as follows:

$$\widehat{\mathbf{\Psi}} = \{\mathbf{f}_j^\intercal \mid \mathbf{f}_j^\intercal \in \mathbf{F}^\intercal \text{ and } j \in \mathbf{S}^*\} = \{\widehat{\boldsymbol{\psi}_1}, \widehat{\boldsymbol{\psi}_2}, \ldots, \widehat{\boldsymbol{\psi}_k}, \ldots \widehat{\boldsymbol{\psi}_q}\} \in \mathbb{R}^{m\times q} \tag{3.9}$$

where q is the number of snapshots in the brain image $\mathbf{F}$, and $\widehat{\boldsymbol{\psi}_k} \in \mathbb{R}^m$ denotes the snapshot for $k\text{-}th$ stimulus. These selected snapshots are employed in next section for extracting features of the neural activities. Algorithm 3.3 illustrates the whole of procedure for generating the snapshots from the time series $\mathbf{F}$.

3.4 Multi-Region Feature Extraction

In this section, the feature extraction is applied in three steps, i.e., normalizing snapshots to standard space, segmenting the snapshots in the form of automatically detected regions, and removing noise by Gaussian smoothing in the level of each region. As mentioned before, normalizing brain image to the standard space can increase the time and space complexities and decrease the robustness of fMRI analysis, especially in voxel-based methods [1]. On the one hand, most of the previous studies [24–26, 78] preferred to use original datasets instead of the standard version because of the

```
Algorithm 3.3 The Snapshots Selection Algorithm
Input: fMRI time series $\mathbf{F}$, time points (onsets) $\mathbf{S}$, HRF signal $\mathbf{H}$, , Gaussian Parameter $\sigma_G$:
Output: Snapshots $\boldsymbol{\Psi}$, the sets of correlations $\widehat{\boldsymbol{\beta}}$:
Method:
1. Generating the design matrix $\mathbf{D} = \mathbf{S} * \mathbf{H}$.
2. Defining $\mathbf{F} = \mathbf{D}\widehat{\boldsymbol{\beta}} + \varepsilon$.
3. Calculating $\widehat{\boldsymbol{\beta}}$ by using (3.2).
4. Generating Gaussian kernel by (3.4).
5. Smoothing the design matrix by (3.5).
6. Finding locations of the snapshots by (3.8).
7. Calculating snapshots $\widehat{\boldsymbol{\Psi}}$ by using (3.9).
```

mentioned problem. On the other hand, this mapping can provide a normalized view for combing homogeneous datasets. As a result, it can significantly reduce the cost of brain studies and rapidly increase the chance of understanding how the brain works. Employing brain snapshots rather than analyzing whole of data can solve the normalization problem.

Normalization can be formulated as a mapping problem. Indeed, brain snapshots are mapped from $\mathbb{R}^m$ space to the standard space $\mathbb{R}^n$ by using a transformation matrix for each snapshot. There is also another trick for improving the performance of this procedure. Since the set $\widehat{\boldsymbol{\beta}}_i$ denotes the locations of all active voxels for the $i\text{-}th$ category, it represents the brain mask for that category and can be used for generating the transform matrix related to all snapshots belong to that category. For instance, in the example of the previous section, instead of calculating 7 transform matrices for 7 stimuli, we calculate 2 matrices, including one for the category of cats and the second one for the category of houses. This mapping can be denoted as follows:

$$\mathbf{T}_i: \qquad \widehat{\boldsymbol{\beta}}_i \in \mathbb{R}^m \quad \rightarrow \quad \boldsymbol{\beta}_i \in \mathbb{R}^n \tag{3.10}$$

where $\mathbf{T}_i \in \mathbb{R}^{m \times n}$ denotes the transform matrix, $\boldsymbol{\beta}_i = \big((\widehat{\boldsymbol{\beta}}_i)^\intercal \mathbf{T}_i\big)^\intercal$ is the set of correlations in the standard space for $i\text{-}th$ category. This section utilizes the FLIRT algorithm [95] for calculating the transform matrix, which minimizes the following objective function:

$$\mathbf{T}_i = \arg\min(NMI(\widehat{\boldsymbol{\beta}}_i, \mathbf{Ref})) \tag{3.11}$$

where the function NMI denotes the Normalized Mutual Information between two images [95], and **Ref** $\in \mathbb{R}^n$ is the reference image in the standard space. This image must contain the structures of the human brain, i.e. white matter, gray matter, and CSF. These structures can improve the performance of mapping between the brain mask in the selected snapshot and the general form of a standard brain. In addition, the sets of correlations for all of categories in the standard space is denoted by $\boldsymbol{\beta} = \{\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_i, \ldots, \boldsymbol{\beta}_p\} \in \mathbb{R}^{n\times p}$, and the sets of transform matrices is defined by $\mathbf{T} = \{\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_i, \ldots, \mathbf{T}_p\}$. Now, the $Select$ function is denoted as follows to find suitable transform matrix for each snapshot:

$$\begin{aligned}\left(\mathbf{T}^*_j, \boldsymbol{\beta}^*_j\right) = Select(\widehat{\boldsymbol{\psi}_j}, \mathbf{T}, \boldsymbol{\beta}) = \{(\mathbf{T}_i, \boldsymbol{\beta}_i) \mid \mathbf{T}_i \in \mathbf{T}, \boldsymbol{\beta}_i \in \boldsymbol{\beta}\ ,\\ \widehat{\boldsymbol{\psi}_j} \text{ is belonged to the } i-th \text{ category } \implies \widehat{\boldsymbol{\psi}_j} \propto \boldsymbol{\beta}_i \propto \mathbf{T}_i\}\end{aligned} \tag{3.12}$$

where $\mathbf{T}^*_j \in \mathbb{R}^{m\times n}$ and $\boldsymbol{\beta}^*_j \in \mathbb{R}^n$ are the transform matrix and the set of correlations related to the j-th snapshot, respectively. Based on (3.12), each normalized snapshot in the standard space is defined as follows:

$$\mathbf{T}^*_j\colon \widehat{\boldsymbol{\psi}}_j \in \mathbb{R}^m \to \boldsymbol{\psi}_j \in \mathbb{R}^n \implies \boldsymbol{\psi}_j = \left(\left(\widehat{\boldsymbol{\psi}}_j\right)^\intercal \mathbf{T}^*_j\right)^\intercal \tag{3.13}$$

where $\boldsymbol{\psi}_j \in \mathbb{R}^n$ is the j-th snapshot in the standard space. Further, all snapshots in the standard space can be defined by $\boldsymbol{\Psi} = \{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots, \boldsymbol{\psi}_j, \ldots, \boldsymbol{\psi}_q\} \in \mathbb{R}^{n\times q}$. As mentioned before, nonzero values in the correlation sets depict the location of the active voxels. Based on (3.12), this section uses these correlation sets as weights for each snapshot as follows:

$$\boldsymbol{\Theta}_j = \boldsymbol{\psi}_j \circ \boldsymbol{\beta}^*_j \tag{3.14}$$

where $\circ$ denotes Hadamard product, and $\boldsymbol{\Theta}_j \in \mathbb{R}^n$ is the j-th modified snapshot, where the values of deactivated voxels (and also deactivated anatomical regions) are zero in this snapshot. As the final product of normalization procedure, the set of snapshots can be denoted by $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \ldots, \boldsymbol{\Theta}_j, \ldots, \boldsymbol{\Theta}_q\}$. Further, each snapshot can be defined in the voxel level as follows, where $\boldsymbol{\theta}^k_j$ is the k-th voxel of j-th snapshot:

$$\boldsymbol{\Theta}_j = \left[\boldsymbol{\theta}^1_j, \boldsymbol{\theta}^2_j, \ldots, \boldsymbol{\theta}^k_j, \ldots, \boldsymbol{\theta}^n_j\right] \tag{3.15}$$

The next step is segmenting the snapshots in the form of automatically detected regions. Now, consider anatomical atlas $\mathbf{A} \in \mathbb{R}^n = \{\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_\ell, \ldots, \mathbf{A}_L\}$, where $\cap^L_{\ell=1}\{\mathbf{A}_\ell\} = \emptyset$, $\cup^L_{\ell=1}\{\mathbf{A}_\ell\} =$

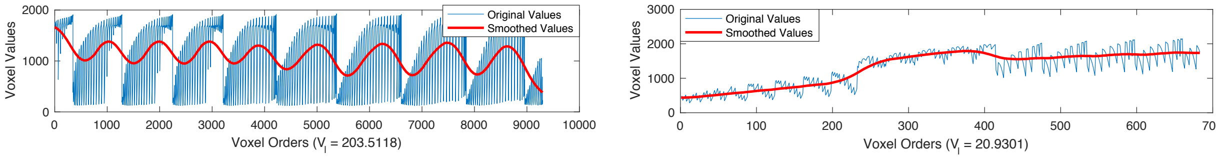


Figure 3.2 Examples of smoothed anatomical regions ($\mathbf{X}_{(j,\ell)}$) in the voxel level

$\mathbf{A}$, and L is the number of all regions in the anatomical atlas. Here, $\mathbf{A}_\ell$ denotes the set of voxel locations in the snapshots for the ℓ-th anatomical region. A segmented snapshot based on the ℓ-th region can be denoted as follows:

$$\boldsymbol{\Theta}_{(j,\ell)} = \{\boldsymbol{\theta}_j^k \mid \boldsymbol{\theta}_j^k \in \boldsymbol{\Theta}_j \text{ and } k \in \mathbf{A}_\ell\} \tag{3.16}$$

where $\boldsymbol{\Theta}_{(j,\ell)} \subset \boldsymbol{\Theta}_j$ is the subset of voxels in the snapshot $\boldsymbol{\Theta}_j$, which these voxels are belonged to the the ℓ-th anatomical region. In addition, the sets of all anatomical regions in the j-th snapshot can be defined by $\boldsymbol{\Theta}_j = \{\boldsymbol{\Theta}_{(j,1)} \cup \boldsymbol{\Theta}_{(j,2)} \cup \dots \cup \boldsymbol{\Theta}_{(j,\ell)} \cup \dots \cup \boldsymbol{\Theta}_{(j,L)}\} = \left[\boldsymbol{\theta}_j^1, \boldsymbol{\theta}_j^2, \dots, \boldsymbol{\theta}_j^k, \dots, \boldsymbol{\theta}_j^n\right]$. The automatically detected active regions can be also defined as follows:

$$\boldsymbol{\Theta}_j^* = \left\{\boldsymbol{\Theta}_{(j,\ell)} | \boldsymbol{\Theta}_{(j,\ell)} \subset \boldsymbol{\Theta}_j \text{ and } \sum_{\boldsymbol{\theta}_{(j,\ell)}^k \in \boldsymbol{\Theta}_{(j,\ell)}} |\boldsymbol{\theta}_{(j,\ell)}^k| \neq 0\right\} \tag{3.17}$$

where $\sum_{\boldsymbol{\theta}_{(j,\ell)}^k \in \boldsymbol{\Theta}_{(j,\ell)}} |\boldsymbol{\theta}_{(j,\ell)}^k|$ represents sum of all voxels in the $\boldsymbol{\Theta}_{(j,\ell)}$. Based on (3.17), active regions in the j-th snapshot can be defined as the regions with non-zero voxels because values of all deactivated voxels are changed to zero by using (3.14). The last step is removing noise by Gaussian smoothing in the level of anatomical regions. As the first step, a Gaussian kernel for each anatomical region can be defined as follows:

$$\sigma_\ell = \frac{N_\ell^2}{5N_\ell^2 \log N_\ell}$$
$$\widehat{\mathbf{V}}_\ell = \left\{ \exp\left(\frac{-\widehat{\mathbf{v}}^2}{2\sigma_\ell}\right) \middle| \widehat{\mathbf{v}} \in \mathbb{Z} \text{ and } -2\lceil\sigma_\ell\rceil \leq \widehat{\mathbf{v}} \leq 2\lceil\sigma_\ell\rceil \right\} \tag{3.18}$$
$$\mathbf{V}_\ell = \frac{\widehat{\mathbf{V}_\ell}}{\sum_j \widehat{\mathbf{v}}_j}$$

where N_ℓ denotes the number of voxels in ℓ-th region, and $\sum_j \widehat{\mathbf{v}}_j$ is sum of all values in the interval $\widehat{\mathbf{V}}_\ell$. Indeed, the level of smoothness is related to σ_ℓ, which is heuristically calculated for each region

Algorithm 3.4 Multi-Region Feature Extraction Algorithm

```
Input: Snapshots $\mathbf{\Psi}$, correlations $\widehat{\boldsymbol{\beta}}$, Ref image, Atlas $\mathbf{A}$:
Output: Smoothed snapshots $\mathbf{X}$:
Method:
1. For each $\widehat{\boldsymbol{\beta}}_i$, calculate transform matrix by (3.11).
2. Mapping $\widehat{\boldsymbol{\psi}}_j$ to standard space by $\mathbf{T}_j^*$ and (3.13).
3. Detecting active voxels for each snapshot by (3.14).
4. Segmenting each snapshot by (3.16).
5. Finding active regions for each snapshot by (3.17).
6. Generating Gaussian kernel by (3.18).
7. Smoothing snapshots by (3.19).
```

based on the number of voxels in that region. As the second step, the smoothed version of the $j\text{-}th$ snapshot can be defined as follows:

$$\begin{gathered} \forall \ell = L1 \ldots L2 \rightarrow \mathbf{X}^{(j,\ell)} = \mathbf{\Theta}_{(j,\ell)} * \mathbf{V}_{\ell}, \\ \mathbf{X}^{(j)} = \{\mathbf{X}^{(j,L1)}, \ldots, \mathbf{X}^{(j,\ell)}, \ldots \mathbf{X}^{(j,L2)}\} \end{gathered} \tag{3.19}$$

where $\mathbf{\Theta}_{(j,\ell)} \in \mathbf{\Theta}_j^*$ is the $\ell\text{-}th$ active region of $j\text{-}th$ snapshot, and $*$ denotes the convolution between the active region and the Gaussian kernel related to that region. Further, $L1$ and $L2$ are the first and the last active regions in the snapshot, where $1 \leq L1 \leq L2 \leq L$. Figure 3.2 demonstrates two examples of smoothed anatomical regions in the voxel level. All smoothed snapshots can be defined by $\mathbf{X} = \{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(j)}, \ldots, \mathbf{X}^{(q)}\}$. Moreover, Algorithm 3.4 shows the whole of procedure for extracting features.

3.5 Weighted Spectral Cluster Ensemble (WSCE)

In this section, we develop an unsupervised cluster ensemble approach to compare the similarities or distances among neural activities of different categories of stimuli. As Figure 3.3 depicted, the extracted neural activities belong to each anatomical region ($\mathbf{X}^{(j,\ell)}, j = 1{:}q$) are employed to apply a base clustering algorithm. Then, the final partitioning is generated by using the results of these algorithms. Indeed, the proposed method firstly generates $\pi^{(\ell)} \in \{1, \ldots, p\}^q, \ell \in [L1, L2]$ as the base clustering results and then combines these results to partition neural activities to p categories of

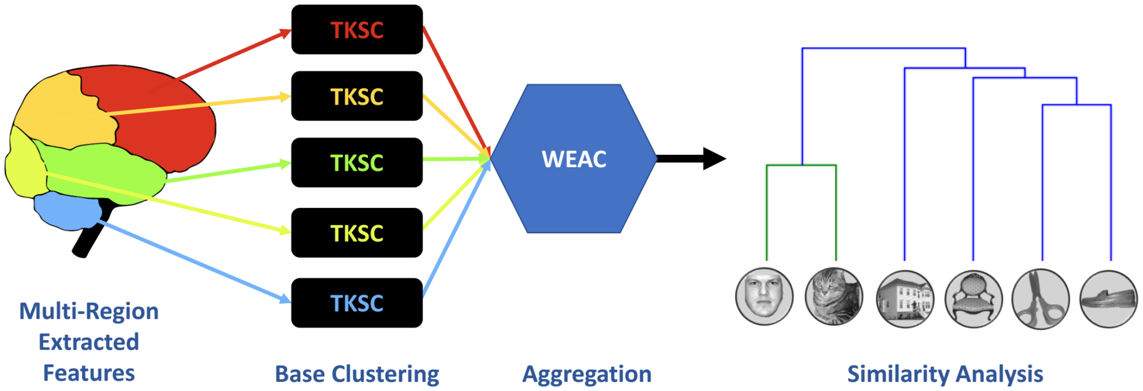


Figure 3.3 Weighted Spectral Cluster Ensemble (WSCE)

stimuli. Here, $\pi^{(\ell)}$ is the ℓ-th base result related to the ℓ-th anatomical region in the reference set ($\mathbf{\Pi} = \{\pi^{(\ell)}\}$, $\ell \in [L1, L2]$).

3.5.1 Two Kernels Spectral Clustering (TKSC)

Like other spectral methods, this section calculates the non-symmetric distances (adjacency) matrix of the neural activities as follows: [96, 97].

$$\mathbf{\Delta}_{i,j}^{(\ell)} = \begin{cases} exp\left(\frac{-\|\mathbf{X}^{(i,\ell)} - \mathbf{X}^{(j,\ell)}\|_2}{\aleph^2}\right) & if\, i \neq j \\ 0 & if\, i = j \end{cases} \tag{3.20}$$

where $\|\mathbf{X}^{(i,\ell)} - \mathbf{X}^{(j,\ell)}\|_2$ will be calculated by Euclidean distance. Indeed, (3.20) can optimize the memory usage [96, 97]. The scaling parameter $\aleph$ controls how rapidly affinity $\mathbf{\Delta}_{i,j}^{(\ell)}$ falls off with the distance between the data points. This thesis uses Ng et al. method for estimating this value automatically (count non-zero values in each columns of the distance matrix) [96, 97].

This section introduces Two Kernels Spectral Clustering (TKSC) algorithm, which can generate all base results ($\mathbf{\Pi}$). Unlike normal clustering algorithms, which just generate a partition as the clustering result, the TKSC algorithm generates two independent consequences, which are called Partitional result and Modular result, for each of the individual clustering results by using two kernels ($\pi^{(\ell)} = \{\widetilde{\pi}_{PA}^{(\ell)}, \widetilde{\pi}_{MO}^{(\ell)}\}$). The Partitional result ($\widetilde{\pi}_{PA}^{(\ell)}$) is a partitioning of data points same as the result of other clustering methods; and the Modular result ($\widetilde{\pi}_{MO}^{(\ell)}$) is a network of data points, which can be represented by a graph. This thesis uses the Modular result as a reference for evaluating the diversity of generated partition by using community detection methods [98, 99]. Furthermore, the kernel in the TKSC refers to Laplacian equation in spectral methods because it transforms data points in the new environment, especially linear environment for non-linear datasets.

Partitional Kernel: This section uses the following equation for generating Partitional result:

$$\mathbf{\Lambda}_{PA}^{(\ell)} = \mathbf{I} - \left(\widetilde{\mathbf{\Delta}}^{(\ell)}\right)^{1/2} \mathbf{\Delta}^{(\ell)} \left(\widetilde{\mathbf{\Delta}}^{(\ell)}\right)^{1/2} \tag{3.21}$$

where $\mathbf{I}$ is the identity matrix [96]; the diagonal matrix $\widetilde{\mathbf{\Delta}}^{(\ell)}$ is calculated as follows:

$$\widehat{\mathbf{\Delta}}^{(\ell)} = \left(\mathbf{\Delta}^{(\ell)} \mathbf{1}_q + 10^{-10}\right)^{-1/2} \tag{3.22}$$

$$\widetilde{\mathbf{\Delta}}_{i,j}^{(\ell)} = \begin{cases} \widehat{\mathbf{\Delta}}_{i,j}^{(\ell)} & if\, i = j \\ 0 & if\, i \neq j \end{cases} \tag{3.23}$$

where ones matrix $\mathbf{1}_q \in 1^{q \times q}$. Now, the eigendecomposition is performed for calculating eigenvectors of $\mathbf{\Lambda}_{PA}^{(\ell)}$:

$$\mathbf{E}^{(\ell)} = eigens\left(\mathbf{\Lambda}_{PA}^{(\ell)}\right) \tag{3.24}$$

where the matrix $\mathbf{E}^{(\ell)}$ is the eigenvectors of Partitional Kernel. The coefficient $\widetilde{\mathbf{E}}^{(\ell)}$ will be defined for normalizing the matrix $\mathbf{E}^{(\ell)}$:

$$\widetilde{\mathbf{E}}_i^{(\ell)} = \left(\sum_{i=1}^{q} \mathbf{E}_{i1}^{(\ell)} \times \mathbf{E}_{i2}^{(\ell)}\right)^{\frac{1}{2}} + 10^{-20} \tag{3.25}$$

where $\mathbf{E}_{ij}^{(\ell)}$ shows the i-th row and j-th column of the matrix $\mathbf{E}^{(\ell)}$; and 10^{-20} is used for omitting the effect of zeros in the matrix $\mathbf{E}^{(\ell)}$. Also, q denotes the number of snapshots ($\widetilde{\mathbf{E}}^{(\ell)} \in \mathbb{R}^q$). The normalized matrix of eigenvectors will be calculated as follows:

$$\widehat{e}_{ij}^{(\ell)} = e_{ij}^{(\ell)} \widetilde{\mathbf{E}}_i^{(\ell)} \tag{3.26}$$

where $\widehat{e}_{ij}^{(\ell)}$ and $e_{ij}^{(\ell)}$ denote the cell located in the i-th row and j-th column of matrices $\widehat{\mathbf{E}}^{(\ell)}$ and $\mathbf{E}^{(\ell)}$, respectively; and $\widetilde{\mathbf{E}}_i^{(\ell)}$ is the i-th row of the vector $\widetilde{\mathbf{E}}^{(\ell)}$ which is used for normalization. The Partitional result of TKSC will be calculated by applying the simple k-means [36–38] on the matrix $\widehat{\mathbf{E}}^{(\ell)}$ as follows:

$$\widetilde{\pi}_{PA}^{(\ell)} = kmeans(\widehat{\mathbf{E}}^{(\ell)}, p) \tag{3.27}$$

Modular Kernel: This section uses the following equation for generating Modular result:

$$\widetilde{\pi}_{MO}^{(\ell)} = \frac{1}{\max\left(\widetilde{\mathbf{\Delta}}^{(\ell)} - \mathbf{\Delta}^{(\ell)}\right)} \widetilde{\mathbf{\Delta}}^{(\ell)} - \mathbf{\Delta}^{(\ell)} \tag{3.28}$$

Algorithm 3.5 Two Kernels Spectral Clustering (TKSC)

```
Input: Neural activities $\mathbf{X}^{(*,\ell)}$, Number of clusters $p$
Output: Partitional result $\widetilde{\pi}_{PA}^{(\ell)}$, Modular result $\widetilde{\pi}_{MO}^{(\ell)}$
Method:
1. Generate similarity matrix $\boldsymbol{\Delta}^{(\ell)}$ by using $\mathbf{X}^{(*,\ell)}$ on (3.20).
2. Generate diagonal matrix $\widetilde{\boldsymbol{\Delta}}^{(\ell)}$ by using $\boldsymbol{\Delta}^{(\ell)}$.
3. $\boldsymbol{\Lambda}_{PA}^{(\ell)}$ by applying $\boldsymbol{\Delta}^{(\ell)}$ and $\widetilde{\boldsymbol{\Delta}}^{(\ell)}$ on (3.21).
4. Generate the matrix $\mathbf{E}^{(\ell)}$ as eigenvectors of $\boldsymbol{\Lambda}_{PA}^{(\ell)}$.
5. Generate $\widehat{\mathbf{E}}_{ij}^{(\ell)}$ as normalized $\mathbf{E}^{(\ell)}$ by using (3.26).
6. Generate $\widetilde{\pi}_{MO}^{(\ell)}$ by applying $\boldsymbol{\Delta}^{(\ell)}$ and $\widetilde{\boldsymbol{\Delta}}^{(\ell)}$ on (3.28).
7. $\widetilde{\pi}_{PA}^{(\ell)} = kmeans(\widehat{\mathbf{E}}^{(\ell)}, p)$
8. Return $\widetilde{\pi}_{PA}^{(\ell)}$ and $\widetilde{\pi}_{MO}^{(\ell)}$
```

where the function max finds the biggest value in the matrix $\widetilde{\boldsymbol{\Delta}}^{(\ell)} - \boldsymbol{\Delta}^{(\ell)}$. Here, we consider $\widetilde{\pi}_{MO}^{(\ell)}$ is an adjacency matrix of graph representation of the neural activities belogn to ℓ-th region. Further, all values in the matrix $\boldsymbol{\Lambda}_{MO}^{(\ell)}$, which is called Modular result, are between zero and one. Algorithm 3.5 shows the pseudo code of the TKSC method.

3.5.2 Weighted Evidence Accumulation Clustering (WEAC)

Tracing errors can control similarity and repetition of specific answers in clustering problems. There is a wide range of metrics, which are based on Shannon's entropy[37, 38], for evaluating the diversity of individual results in the ensemble methods, such as MI [59], NMI [61], APMM [63], MAX [64], and SMI [67]. Shannon's entropy uses the logarithm of the probability of individual results for evaluating the diversity, but there is no mathematical proof that all real-world datasets have logarithmic behavior. In community detection [98, 99], Modularity, which is based on Expected Value, was proposed for solving this problem. Recently, many papers proved that modularity [98, 99] could estimate the diversity on graph-based datasets such as brain networks better than entropy-based methods. Unfortunately, modularity can measure the diversity only for graph data [98]. This thesis proposes TKSC, which can generate a graph based result, called Modular result, for any types of datasets in the real-world application. Since modularity was defined for community detection area, this thesis introduces a redefined version of modularity metric for general clustering problems, which is called Normalized Modularity (NM). It is used for evaluating the diversity of the individual results based on the Modular result of the TKSC as follows:

$$\mathbf{O} = NM(\mathbf{P}, \mathbf{M}) = \frac{1}{2} + \frac{1}{4\sum_{\mathbf{M}} \mathbf{M}_{ij}} \sum_{ij} \left[\Gamma(\mathbf{M}_{ij}) - \frac{\delta(\mathbf{M}_i)\delta(\mathbf{M}_j)}{2\sum_{\mathbf{M}} \mathbf{M}_{ij}} \right] (1 - \Gamma(\mathbf{P}_i - \mathbf{P}_j)) \tag{3.29}$$

$$\Gamma(\mathbf{x}) = \begin{cases} 0 & \mathit{if}\, \mathbf{x} = 0 \\ 1 & \mathit{Otherwise} \end{cases} \tag{3.30}$$

where **P** is a Partitional result, **M** denotes a Modular results, $\delta(\mathbf{M}_\ell)$ shows the degree of ℓ-th node in the graph of matrix **M** (how many rows contains non-zero value in the columns ℓ), $\Gamma(\mathbf{x})$ is zero if and only if vector **x** is a zero vector. Indeed, we must calculate $\mathbf{O}^{(\ell)} = NM(\widetilde{\pi}_{PA}^{(\ell)}, \widetilde{\pi}_{MO}^{(\ell)})$, $\ell \in [L1, L2]$ for all active regions, where it is always $0 \le NM \le 1$. Now, we develop Weighted Evidence Accumulation Clustering (WEAC) for generating the co-association matrix:

$$\zeta_{ij} = \frac{\sum_{\ell=L1}^{L2} \mathbf{O}_i^{(\ell)} + \mathbf{O}_j^{(\ell)}}{2q(L2 - L1)} \tag{3.31}$$

where $\mathbf{O}_i^{(\ell)}$ illustrates the Normalized Modularity related to ℓ-th anatomical region and i-th snapshot. Further, the final co-association matrix, which is a symmetric matrix, will be generated as follows:

$$\xi = WEAC(\mathbf{\Pi}) = \begin{pmatrix} \zeta_{11} & \zeta_{12} & \cdots & \zeta_{1q} \\ \zeta_{21} & \zeta_{22} & \cdots & \zeta_{2q} \\ \vdots & \vdots & \vdots & \vdots \\ \zeta_{i1} & \zeta_{i2} & \zeta_{ij} & \zeta_{iq} \\ \vdots & \vdots & \vdots & \vdots \\ \zeta_{q1} & \zeta_{q2} & \cdots & \zeta_{qq} \end{pmatrix} \tag{3.32}$$

where q is the number of the snapshots; and ζ_{ij} denotes the final aggregation for i-th and j-th stimuli. Algorithm 3.6 illustrates the pseudo code of the proposed method. In this algorithm, **X** is the neural activities; p is the number of clusters in the final result (i.e., number of stimuli categories). A Euclidean metric also measures the distances. The TKSC function builds the partitions and modules of individual results, and NM function evaluates these results. Then, the evaluated results will be added to reference set (**Π**). The Average-Linkage function creates the final ensemble according to the average linkage method [36–38, 63, 64].

Algorithm 3.6 Weighted Spectral Cluster Ensemble (WSCE)

```
Input: Data points $\mathbf{X}$, Number of clusters $p$
Output: final result $\mathbf{P}_{final}$
Method:
FOR $\ell = L1{:}L2$
    $[\widetilde{\pi}^{(\ell)}_{PA}, \widetilde{\pi}^{(\ell)}_{MO}] = TKSC\left(\mathbf{X}^{(*,\ell)}, p\right)$
    $\mathbf{O}^{(\ell)} = NM(\widetilde{\pi}^{(\ell)}_{PA}, \widetilde{\pi}^{(\ell)}_{MO})$
    Add $\widetilde{\pi}^{(\ell)}_{PA}$, $\widetilde{\pi}^{(\ell)}_{MO}$, and $\mathbf{O}^{(\ell)}$ to the reference set $\mathbf{\Pi}$.
END FOR
Generate co-association matrix $\xi = WEAC(\mathbf{\Pi})$
$\mathbf{P}_{final}$ = Average-Linkage($\xi$)
```

3.6 Multi-Region Ensemble Learning (MREL)

As a classical classification method, Support Vector Machine (SVM) [100, 101] decreases the operating risk and can find an optimized solution by maximizing the margin of error. As a result, it can mostly generate better performance in comparison with other methods, especially for binary classification problems. Therefore, SVM is used in the wide range of studies for creating predictive models [24–26, 78]. As Figure 3.4 depicted, the final goal of this section is employing the $\ell1$-regularization SVM [100] method for creating binary classification at the level of anatomical region, and then combining these classifiers by using the Bagging algorithm [102] for generating the final predictive model.

As mentioned before, fMRI time series for a subject can be denoted by $\mathbf{F}$. Since fMRI experiment is mostly multi-subject, this thesis denotes $\mathbf{F}_u, = 1{:}U$ as fMRI time series (sessions) for all subjects, where U is the number of subjects. In addition, $\tau = \sum_{u=1}^{U} q_u$ is defined as the number of all snapshots in a unique fMRI experiment. Here q_u is the number of snapshots for the u-th subject. Further, the original ground truth (the title of stimuli such that cats, houses, etc.) for all snapshots is denoted by $\mathbf{Y} = \{y_1, y_2, \dots, y_j, \dots y_\tau\}$, where y_j denotes the ground truth for j-th snapshot. Since this thesis uses a one-versus-all strategy, we can consider that $y_j \in \{-1, +1\}$. This thesis applies following objective function on automatically detected active regions as the $\ell1$-regularization SVM method for

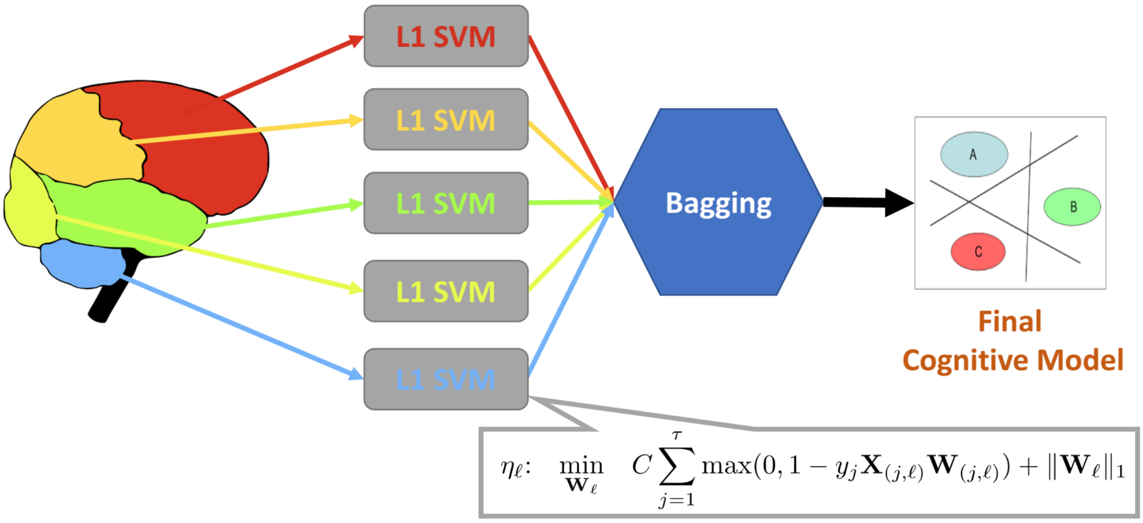

L1 SVM
L1 SVM
L1 SVM
L1 SVM
L1 SVM
Bagging
A
B
C
Final
Cognitive Model
$\eta_\ell\colon \quad \min_{\mathbf{W}_\ell} \quad C \sum_{j=1}^{\tau} \max(0, 1 - y_j \mathbf{X}_{(j,\ell)} \mathbf{W}_{(j,\ell)}) + \|\mathbf{W}_\ell\|_1$

Figure 3.4 Multi-Region Ensemble Learning (MREL)

creating binary classification in the level of ROIs [24, 100]:

$$\eta_\ell\colon \quad \min_{\mathbf{W}_\ell} \quad C \sum_{j=1}^{\tau} \max(0, 1 - y_j \mathbf{X}_{(j,\ell)} \mathbf{W}_{(j,\ell)}) + \|\mathbf{W}_\ell\|_1 \tag{3.33}$$

where $C > 0$ is a real positive number, $\mathbf{X}_{(j,\ell)}$ and y_j denote the voxel values of ℓ-th region and the class label of j-th snapshot, respectively. Further, $\mathbf{W}_\ell = [\mathbf{W}_{(1,\ell)}, \mathbf{W}_{(2,\ell)}, \dots, \mathbf{W}_{(j,\ell)}, \dots, \mathbf{W}_{(\tau,\ell)}]$ is the generated weights for predicting MVP model based on the ℓ-th active region. Here, $\|\mathbf{W}\|_1 = \max_j \sum_i |\mathbf{W}_{ij}|$. The classifier for ℓ-th region is also denoted by η_ℓ, where all of these classifiers can be defined by $\eta = \{\eta_{L1}, \dots, \eta_\ell, \dots \eta_{L2}\}$. The final step in the proposed method is combining all classifiers (η) by Bagging [102] algorithm for generating the MVP final predictive model. Indeed, Bagging method uses the average of predicted results in (3.33) for generating the final result ($\eta_{final} = \sum_{\ell=L1}^{L2} \eta_\ell$) [102, 103]. Algorithm 3.7 shows the whole of procedure in the proposed method by using Leave-One-Out (LOO) cross-validation in the subject level.

3.7 Experiments

The empirical studies will be presented in this section. We employ the Montreal Neurological Institute (MNI) 152 T1 4mm [44, 45] as the reference image (**Ref**) in (3.11) for mapping the extracted snapshots to the standard space ($\widehat{\boldsymbol{\psi}}_i \to \boldsymbol{\psi}_i$). The size of this image in 3D scale is $X = 46, Y = 55, Z = 46$. Moreover, the *Talairach* Atlas [43] (including $L = 1105$ regions) in the standard space is used in (3.17) for extracting features. Further, all of algorithms are implemented in Python 3 on a

Algorithm 3.7 Multi-Region Ensemble Learning (MREL) by using LOO cross validation

```
Input: fMRI time series $\mathbf{F}_u, u = 1{:}U$, Onsets $\mathbf{S}_u, u = 1{:}U$, HRF signal $\mathbf{H}$, Gaussian Parameter $\sigma_G$ (default $\sigma_G = 1$):
Output: MVP performance ($ACC, AUC$)
Method:
FOR $u = 1{:}U$
    Create train set $\mathbf{F}_{Tr} = \{\mathbf{F}_j | j = 1{:}U, j \neq u\}$.
    Extract snapshots of $\mathbf{F}_{Tr}$ by using Algorithm 3.3.
    Generate features of $\mathbf{F}_{Tr}$ by using Algorithm 3.4.
    Train binary classifiers $\eta$ by using $\mathbf{F}_{Tr}$ and (3.33).
    Generate final predictor ($\eta_{final}$) by using Bagging.
    Consider $\mathbf{F}_u$ as test set.
    Extract snapshots for $\mathbf{F}_u$ by using Algorithm 3.3.
    Generate features for $\mathbf{F}_u$ by using Algorithm 3.4.
    Apply test set on the final predictor ($\eta_{final}$).
    Calculate performance of $\mathbf{F}_u$ ($ACC_i, AUC_i$) [103].
END FOR
Accuracy: [103]: $ACC = \sum_{i=1}^{U} ACC_i \big/ U$.
AUC [103]: $AUC = \sum_{i=1}^{U} AUC_i \big/ U$.
```

PC with certain specifications[1] by authors in order to generate experimental results.

3.7.1 Correlation Analysis

The correlations of the extracted features will be compared with the correlations of the original voxels in this section. Previous studies illustrated that patterns of different Abstract-Categories (ACs) [25], which is extracted from a suitable feature representation, must provide distinctive correlation values [25, 82]. Therefore, the main assumption in this section is that better feature representation (extraction) can improve the correlation analysis, where the correlation between different categories of visual stimuli must be significantly smaller than the correlation between stimuli belonged to the same

[1] DEL , CPU = Intel Xeon E5-2630 v3 (8×2.4 GHz), RAM = 64GB, OS = KDE Neon 16.04.4

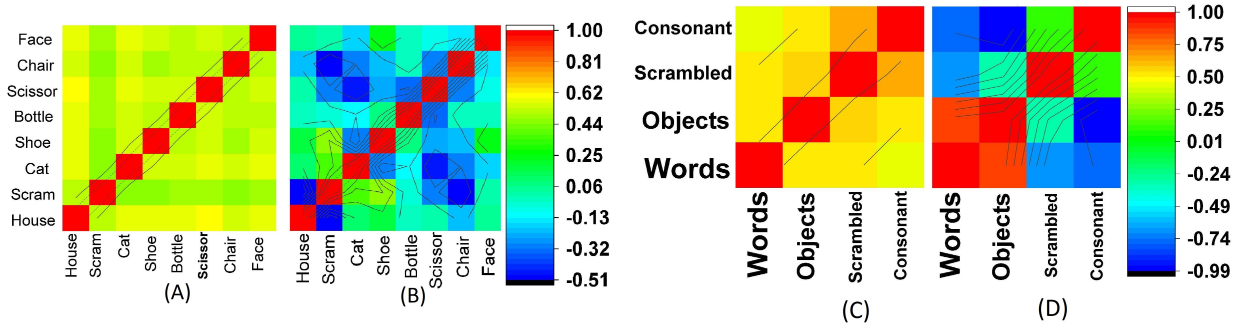


Figure 3.5 Correlation analysis for R105 dataset (A) in the voxel level, (B) feature level, for R107 dataset (C) in the voxel level, (D) feature level.

category. In order to provide a better perspective, the extracted features are compared by considering two different levels. At the first level, the feature space is compared with the whole of raw voxels in the original space, where this comparison analyzes the correlation between whole-brain data and automatically detected anatomical regions. At the second level, the correlation values among different ACs are compared in the feature space that it shows how much the feature space is well-designed.

This section presents two examples to compare the effect of using snapshots rather than employing all time points. Figure 3.5 A, and C respectively demonstrate correlation matrix at the voxel level for the datasets R105, and R107. Further, Figure 3.5 B, and D respectively illustrate the correlation matrix in the feature level for the datasets R105, and R107. Since neural activities are sparse, high-dimensional and noisy in voxel level, it is so hard to discriminate between different categories in Figure 3.5 A, and C. By contrast, Figure 3.5 B, and D provided distinctive and informative representation when the proposed method used the extracted features.

The correlation between different ACs can also be meaningful in the feature space. In R105 and R107, the scramble (nonsense) stimuli have a low correlation in comparison with sensible categories. As another example in R105, human faces are mostly correlated to the photos of cats and houses in comparison with other objects. Another interesting example is the correlation between meaningful stimuli (words and objects) and nonsense stimuli (scrambles and consonants) in R107, where the meaningful stimuli are highly correlated, and their correlations with nonsense stimuli are negative. Indeed, the noisy and sparse raw voxels are not suitable (wise) in order to train a high-performance cognitive model. It is worth noting that we have the same tendency for the reset of datasets. In Chapter 4, we will compare all datasets together.

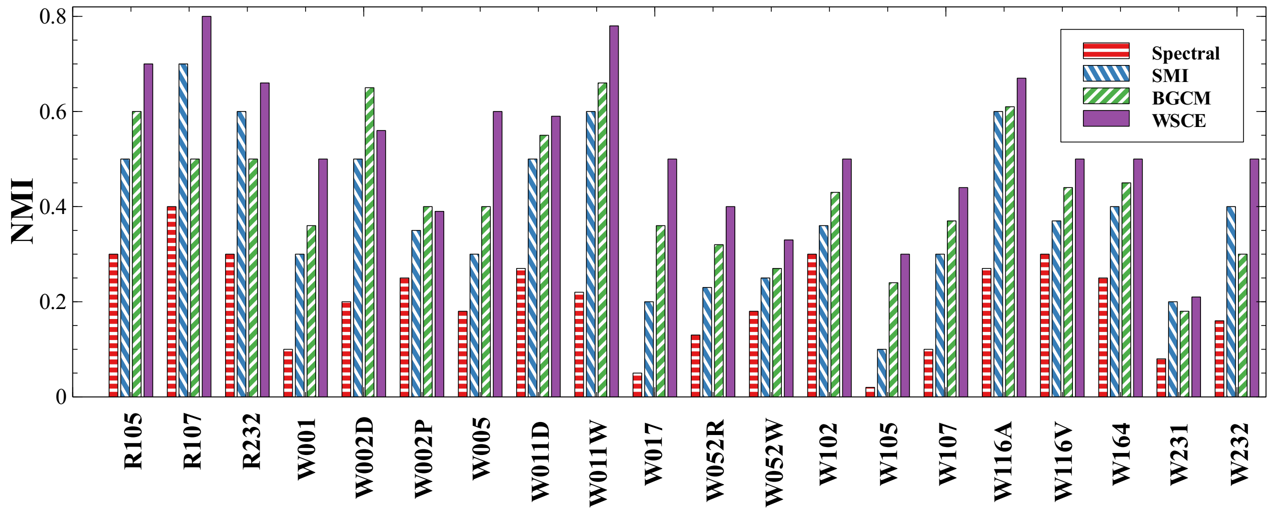


Figure 3.6 Unsupervised Analysis of different neural activities by using WSCE

3.7.2 Unsupervised Analysis

This section analyzes the neural activities by using unsupervised clustering techniques. Figure 3.6 compares unlabeled neural activities by using NMI metric. In this figure, the results of WCES are compared with Spectral clustering [96] as a baseline. Further, the performance of Standardized Mutual Information (SMI) is reported as an NMI-based ensemble technique [67]. As a graph-based ensemble approach, we compare the proposed method with a graph-based consensus maximization (BGCM) method [73]. Here, the Gaussian parameter for smoothing the design matrix is considered $\sigma_G = 1$. Moreover, we apply the neural activities of different snapshots to these methods and then calculate the average of within-cluster similarity by using NMI metric. As Figure 3.6 depicted, WSCE generates better performance in comparison with other methods because it uses the anatomical features of neural activities to improve the clustering analysis. Indeed, WSCE can demonstrate that how much the neural activities in an Abstract-Category (AC) are similar to each other. In Chapter 4, we develop new methods in order to compare between-class similarities or differences by employing supervision information.

3.7.3 Supervised Analysis

The performance of MREL method is compared with prevalent feature selection algorithms, which were proposed for decoding the distinctive stimuli in the human brain. We first compared the proposed approach with ℓ1-SVM as a baseline, where the raw neural activities are directly applied to this algorithm. Further, our method is compared with Principal Component Analysis (PCA) and Independent Component Analysis (ICA) as two unsupervised approaches for feature selection. As a supervised alternative, we also report the performance of Linear Discriminant Analysis (LDA) is indicated. Here, we firstly select snapshots by using the proposed method and then using the same number of time points in the mentioned methods in order to select features. This section compares the performance of the mentioned methods as well as the proposed method by using leave-one-subject-out cross-validation for all dataset except R107 that uses leave-four-subject-out cross-validation. Further, the Gaussian parameter for smoothing the design matrix is considered $\sigma_G = 1$. The effect of different values of this parameter on the performance of the proposed method will be discussed in the next selection. Table 3.1 illustrates the accuracy of classification analysis by using these feature selection techniques, where the proposed method generates better performance in comparison with other feature selection techniques. Indeed, our method uses snapshots with the higher probability of representing neural activities. Thus, it can present better discrimination to apply a classification analysis.

3.7.4 Parameters Analysis

In this section, the effect of different parameters on the performance of the proposed feature selection method will be analyzed. As the first parameter, σ_G in (3.3) is heuristically defined to change the level of smoothness in the design matrix. The general assumption here is the $0 < \sigma_G < 1$ can create design matrix, which is sensitive to small spikes. As a result, the detected local maximums and also the number of snapshots will be more than the real number. Moreover, $\sigma_G > 1$ can rapidly increase the level of smoothness, and also can remove some weak local maximums, especially in the event-related fMRI datasets. Figure 3.7.A illustrates the effect of different σ_G values on the number of wrongs detected snapshots in the region-based datasets. As depicted in this figure, the $\sigma_G = 1$ generated better results in comparison with other values. This is the main reason that this paper uses $\sigma_G = 1$ as the default value in the empirical studies.

The next parameter that can affect the performance of the proposed method is the distance metric in the objective function (3.11) for mapping functional snapshots to the standard space. Figure 3.7.B and C demonstrate two examples of the error of registration (normalization) in the detected snapshots. Here, gray parts show the anatomical atlas, the colored parts (yellow and blue) define the functional activities, and also the red rectangles illustrate the error areas after registration. Indeed, these errors

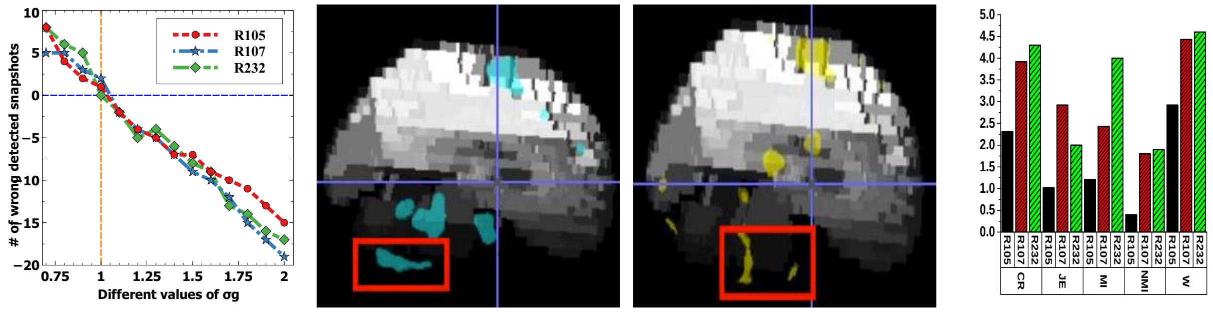
Table 3.1 Accuracy of classification analysis for evaluating different feature selection techniques (max±std)

Datasets	ℓ1-SVM	PCA	ICA	LDA	MREL
R105	16.72±1.89	19.04±0.34	21.76±0.30	**35.20±0.38**	34.99±0.54
R107	26.39±2.60	28.06±0.45	30.47±0.07	32.04±0.07	**40.27±0.93**
R232	30.65±0.79	33.34±1.43	36.45±0.28	40.69±0.16	**52.49±0.07**
W001	25.47±0.36	29.39±0.14	28.54±0.17	33.07±0.96	**37.19±0.63**
W002D	61.69±0.86	62.41±0.63	65.61±0.13	66.75±0.73	**67.84±0.76**
W002P	64.98±0.15	67.68±0.36	67.34±0.72	70.67±0.53	**73.25±0.12**
W005	32.80±0.93	35.59±0.84	38.59±0.33	40.19±0.73	**45.16±0.59**
W011D	41.26±0.13	63.21±0.13	60.08±0.71	66.08±0.22	**68.81±0.43**
W011W	31.37±0.54	31.97±0.02	32.62±0.63	33.20±0.51	**35.34±0.68**
W017	22.31±0.72	30.27±0.11	30.23±0.50	40.95±0.65	**45.69±0.09**
W052R	54.83±0.17	63.76±0.98	62.32±0.81	68.19±0.13	**70.90±0.51**
W052W	53.42±0.99	60.01±0.90	58.83±0.46	63.84±0.16	**67.76±0.30**
W102	52.33±0.46	53.93±0.45	54.91±0.30	58.19±0.11	**71.93±0.25**
W105	17.72±0.10	22.62±0.05	22.65±0.93	**38.68±0.08**	37.30±0.48
W107	32.24±1.61	32.58±0.76	38.42±0.89	45.73±0.82	**51.16±0.87**
W116A	56.46±0.19	61.65±0.86	61.86±0.69	64.94±0.73	**65.98±0.95**
W116V	58.03±0.45	61.45±0.99	63.97±0.24	65.21±0.96	**68.32±0.32**
W164	50.82±0.15	62.06±0.26	60.23±0.07	67.05±0.35	**71.86±0.38**
W231	27.66±0.92	45.34±0.49	43.65±0.15	55.38±0.84	**63.77±0.55**
W232	29.97±0.46	31.68±0.22	30.25±0.57	32.41±0.49	**37.63±0.26**

can be formulated as the nonzero areas in the snapshots which are located in the zero area of the anatomical atlas (the area without region number). The performance of objective function (3.11) on region-based datasets is analyzed in Figure 3.7.D by using different distance metrics, i.e. Woods function (W), Correlation Ratio (CR), Joint Entropy (JE), Mutual Information (MI), and Normalized Mutual Information (NMI) [61, 95]. As depicted in this figure, the NMI generated better results in comparison with other metrics.

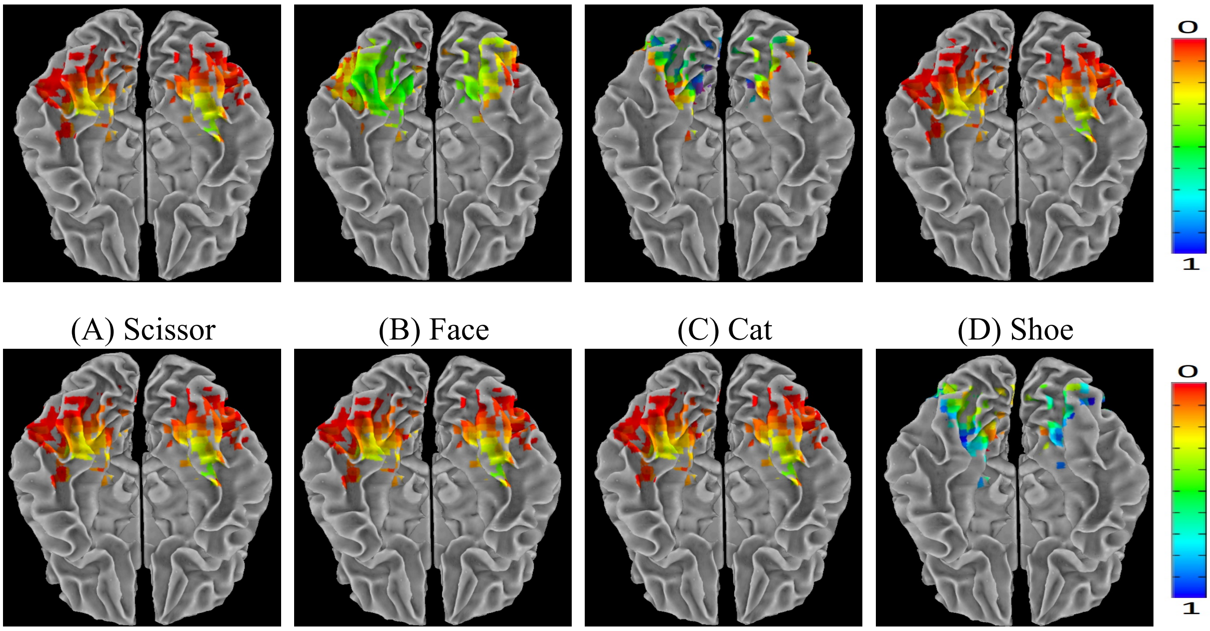
3.7.5 Representing Neural Activities

In this section, we visualize some samples of neural activities that are selected as the snapshots in data R105. As Figure 3.8 depicted, while some of these neural activities are significantly distinctive,



(A) (B) (C) (D)

Figure 3.7 Parameters Analysis: (A) The effect of different σ_G values on the # of wrong detected snapshots, (B) and (C) two examples for the error of registration (normalization): the red rectangles illustrate the error areas after registration, (D) The effect of different objective functions in (3.11) on the error of registration.

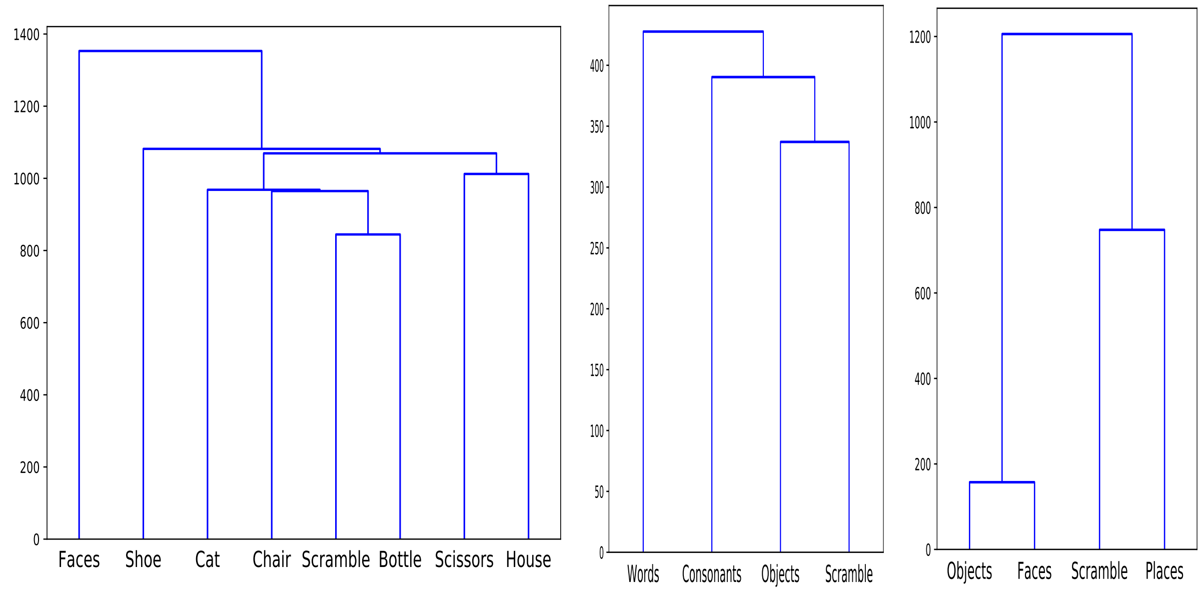


(A) Scissor (B) Face (C) Cat (D) Shoe

(E) House (F) Scramble (G) Bottle (H) Chair

Figure 3.8 Representing neural activities of selected snapshots in data R105

the rest of them are highly correlated. However, the proposed feature selection technique segment these neural activities to different anatomical regions, and then we select active regions. Thus, the sparsity of these features is significantly reduced in within region. Indeed, this is the main motivation for us to analyze the features in the anatomical region level rather than the whole of data at same time.



(A) R105

(B) R107

(C) R232

Figure 3.9 Unsupervised similarity analysis based on WSCE

3.7.6 Visualizing Unsupervised Similarity

In this section, we visualize the unsupervised similarity of the region-based datasets by using the generated WEAC matrix in the WSCE method. Figure 3.9 illustrates the dendrogram of the neural activities, where similar categories of stimuli are connected to each other in a nested structure. It is worth noting that these similarities are directly generated based on the neural activities without any supervised information. Thus, there are suitable to estimate how much the neural pattern in different categories of stimuli are linearly independent (or dependent) to each other. In Chapter 4, we will present a novel supervised approach that is called Deep Representational Similarity Analysis (DRSA) for analyzing the similarity of the neural activities. Indeed, DRSA not only uses the brain patterns to apply the similarity analysis but also it employs the supervised information in the design matrix to detect within- or between- class stimuli.

3.8 Conclusion

There is a wide range of challenges for fMRI analysis. In this section, we discuss some of them, i.e., decreasing noise and sparsity, defining effective regions of interest (ROIs), visualizing results, and the cost of brain studies. In overcoming these challenges, this section proposes Multi-Region

Neural Representation as a novel feature space for decoding different stimuli in the human brain. The proposed method is applied in three stages: firstly, snapshots of brain image (each snapshot represents neural activities for a unique stimulus) are selected by finding local maximums in the smoothed version of the design matrix. Then, features are generated in three steps, including normalizing to standard space, segmenting the snapshots in the form of automatically detected anatomical regions, and removing noise by Gaussian smoothing in the level of the detected regions. Finally, we propose two learning approaches. Indeed, extracted features can be analyzed by using both unsupervised learning and supervised learning. This thesis proposed a cluster ensemble approach in order to apply unsupervised learning, where similarities or distances between neural activities can be compared across subjects. As the supervised alternative, we develop an ensemble classification (i.e., bagging technique) on binary $\ell 1$-regularized SVM classifiers, where they are created by employing each of neural activities in the level of ROIs, i.e., each snapshot represents neural activities for a unique stimulus. Experimental studies show the superiority of our proposed method in comparison with state-of-the-art methods. In addition, the time complexity of the proposed method is naturally lower than the classical methods because it employs a snapshot of brain image for each stimulus rather than using the whole of time series.

Chapter 4. Deep Representational Similarity Analysis

As one of the fundamental approaches in fMRI analysis, Representational Similarity Analysis (RSA) [7, 27, 28] is a supervised approach that evaluates the similarities (or distances) between distractive cognitive tasks. In practice, RSA can be mathematically formulated as a multi-set (group) regression problem, i.e., a linear model for mapping between the matrix of neural activities and the design matrix [34]. Original RSA employs basic linear approaches, such as Ordinary Least Squares (OLS) [7] or General Linear Model (GLM) [28]. Indeed, these methods cannot provide acceptable performances on real-world datasets, e.g., datasets with broad Region of Interest (ROI) or whole-brain fMRI data [3, 13, 104, 105]. On the one hand, the number of voxels in most of fMRI datasets are more than time points. Thus, the matrix of neural activities may not be full rank [13]. On the other hand, the mentioned methods must calculate the inverse of the covariance matrix of the neural activities for solving the RSA problems [19]. This inverse may reduce the stability of the results when the covariance matrix includes low Signal-to-Noise Ratio (SNR) [3].

As the first group of modern approaches, some of the new RSA methods utilize the Bayesian technique [3, 29]. As one of these algorithms, Bayesian RSA (BRSA) [3] considers the covariance matrix as a hyper-parameter generative model and then calculates this matrix from neural activities. Although Bayesian methods can significantly improve the SNR issue and even handle some nonlinear datasets, they are limited to a restricted transformation function (Gaussian distributions of the hyper-parameters). As another problem in the classical approaches, OLS and GLM also do not use the regularization term to avoid the overfitting. The second group of modern approaches focused on the regularization issue. While Ridge Regression method [30] utilizes an additional norm $\ell 2$ for solving the mentioned issue, Least Absolute Shrinkage and Selection Operator (LASSO) method [31] employs norm $\ell 1$ to regularize the regression problem. As another alternative, the Elastic Net method [32] made a trade-off between $\ell 1$ and $\ell 2$ norms. Moreover, other techniques developed novel regularization terms, such as Octagonal Shrinkage and Clustering Algorithm for Regression (OSCAR) [33] or Ordered Weighted $\ell 1$ (OWL) [34, 35]. As the primary problem, these methods always consider that the relations between features are linear.

4.1 Representational Similarity Analysis (RSA)

fMRI time series collected from ℓ-th subject can be denoted by $\mathbf{X}^{(\ell)} = \Big\{x_{ij}\Big\} \in \mathbb{R}^{T\times V_{org}}, 1 \leq i \leq T, 1 \leq j \leq V_{org}$, where T is the number of time points, and V_{org} denotes the number of voxels in the original space. This thesis assumes that the neural activities of each subject are column-wise standardized, i.e., $\mathbf{X}^{(\ell)} \sim \mathcal{N}(0,1)$. We can also consider this condition as a preprocessing step if the original data is not standardized. Indeed, RSA method is looking for the following objective function:

$$\min_{\mathbf{B}^{(\ell)}} \Big|\Big|\mathbf{X}^{(\ell)} - \mathbf{D}^{(\ell)}\mathbf{B}^{(\ell)}\Big|\Big|_F^2 + r(\mathbf{B}^{(\ell)}), \tag{4.1}$$

where $\mathbf{D}^{(\ell)} = \Big\{d_{ik}\Big\} \in \mathbb{R}^{T\times P}, d_{ik} \in \mathbb{R}, 1 \leq i \leq T, 1 \leq k \leq P$ is the design matrix, $\mathbf{B}^{(\ell)} = \Big\{\beta_{kj}\Big\} \in \mathbb{R}^{P\times V_{org}}, \beta_{kj} \in \mathbb{R}, 1 \leq k \leq P, 1 \leq j \leq V_{org}$ denotes the matrix of estimated regressors, and $r(\mathbf{B}^{(\ell)})$ is the regularization term for ℓ-th subject. Here, P denotes the number of distinctive categories of stimuli, and $\mathbf{d}_{.k}^{(\ell)} \in \mathbb{R}^T, 1 \leq k \leq P$ as the k-th column of the design matrix is the convolution of the onsets of k-th category ($\mathbf{o}_{.k}^{(\ell)} \in \mathbb{R}^T$) with $\boldsymbol{\Xi}$ as the Hemodynamic Response Function (HRF) signal, i.e., $\mathbf{d}_{.k}^{(\ell)} = \mathbf{o}_{.k}^{(\ell)} * \boldsymbol{\Xi}$ [3, 19].

In (4.1), the regularization term is zero ($r(\mathbf{B}) = 0$) for non-regularized methods, including OLS, GLM, and BRSA. The term $r(\mathbf{B})$ is $\alpha\Big|\Big|\mathbf{B}\Big|\Big|_F^2$ for Ridge Regression, $\alpha\Big|\Big|\mathbf{B}\Big|\Big|_{1,2}$ for LASSO method, $\alpha\rho\Big|\Big|\mathbf{B}\Big|\Big|_{1,2} + 0.5\alpha(1-\rho)\Big|\Big|\mathbf{B}\Big|\Big|_F^2$ for Elastic Net method. Here, we have $\alpha > 0, 0 < \rho < 1$, and $\Big|\Big|\mathbf{B}\Big|\Big|_{1,2} = \sum_{k=1}^{P} \Big|\Big|\boldsymbol{\beta}_{k.}\Big|\Big|_2$. In addition, OWL utilizes $r(\mathbf{B}) = \sum_{k=1}^{P} \lambda_k \Big|\Big|\boldsymbol{\beta}_{[k].}\Big|\Big|_2$, where the rows of matrix $\mathbf{B}$ is sorted from greatest $\ell 2$ norm to the smallest one, and λ_k is non-negative and non-increasing weights. In some sense, the OWL regularization term is a generalized version of the OSCAR regularization, i.e. λ_k is weights with linear decay [34].

In order to generalize RSA for multi-subject fMRI datasets, we calculate the mean of the regressors matrices across subjects:

$$\mathbf{B}^* = \frac{1}{S}\sum_{\ell=1}^{S} \mathbf{B}^{(\ell)}, \tag{4.2}$$

where S denotes the number of subjects, and each row of $\mathbf{B}^* \in \mathbb{R}^{P\times V_{org}} = \Big\{\boldsymbol{\beta}_{1.}^*, \dots, \boldsymbol{\beta}_{P.}^*\Big\}, \boldsymbol{\beta}_{k.}^* \in \mathbb{R}^{V_{org}}$ illustrates the extracted neural signature belong to k-th category of cognitive tasks.

Three metrics will be used to evaluate the performance of RSA methods. As the first metric, we calculate the mean of square error for analyzing the accuracy of regression:

$$MSE = \frac{1}{TSV}\sum_{\ell=1}^{S}\sum_{i=1}^{T}\sum_{j=1}^{V}\Big(x_{ij}^{(\ell)} - \sum_{k=1}^{P} d_{ik}^{(\ell)}\beta_{kj}^{(\ell)}\Big)^2. \tag{4.3}$$

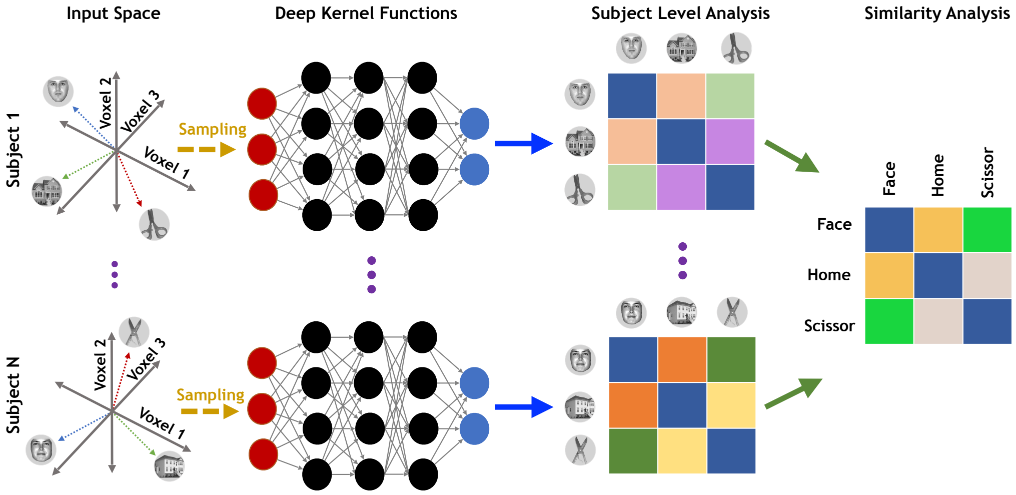


Figure 4.1 Deep Representational Similarity Analysis (DRSA) approach

The next two techniques evaluate between-class correlation and between-class covariance of the regressors matrices:

$$CR = \frac{1}{S}\sum_{\ell=1}^{S} \max_{\substack{1\leq i\leq P,\\ i<j\leq P}} \left\{ Corr(\boldsymbol{\beta}_{i.}^{(\ell)}, \boldsymbol{\beta}_{j.}^{(\ell)}) \right\}, \tag{4.4}$$

$$CV = \frac{1}{S}\sum_{\ell=1}^{S} \max_{\substack{1\leq i\leq P,\\ i<j\leq P}} \left\{ Cov(\boldsymbol{\beta}_{i.}^{(\ell)}, \boldsymbol{\beta}_{j.}^{(\ell)}) \right\}, \tag{4.5}$$

where $\boldsymbol{\beta}_{i.}^{(\ell)}, \boldsymbol{\beta}_{j.}^{(\ell)} \in \mathbf{B}^{(\ell)}$, function $Corr$ is the Pearson correlation, and function Cov calculates the covariance between two vectors. All of these three metrics must be minimized for an ideal solution [3, 34, 105].

4.2 Deep Representational Similarity Analysis (DRSA)

As Figure 4.1 depicted, DRSA maps nonlinear neural activities to a linear embedded space by using a transformation function, i.e. $\mathbf{x} \in \mathbb{R}^{V_{org}} \to f(\mathbf{x}) \in \mathbb{R}^{V}$, where $V \leq V_{org}$ denotes the number of mapped features in the linear embedded space. Although f can be any restricted fixed transformation function (such as Gaussian or Polynomial), this thesis uses multiple stacked layers of nonlinear transformation function as follows:

$$\begin{gathered} f\big(\mathbf{x};\boldsymbol{\theta}\big) = \mathbf{W}_C\mathbf{h}_{C-1} + \mathbf{b}_C, \\ \mathbf{h}_m = g(\mathbf{W}_m\mathbf{h}_{m-1} + \mathbf{b}_m) \text{ for } 2 \leq m < C, \end{gathered} \tag{4.6}$$

where $\mathbf{h}_1 = \mathbf{x}$, $C \geq 3$ is the number of deep network layers, $\boldsymbol{\theta} = \big\{\mathbf{W}_m, \mathbf{b}_m \text{ for } 2 \leq m \leq C\big\}$ denotes all network parameters, and g is a nonlinear function applied componentwise, i.e., Rectified Linear Unit (ReLU), sigmoid, or $tanh$ [13]. By considering $U^{(m)}, 2 \leq m < C$ as the number of units in $m\text{-}th$ intermediate layer, the parameters of the output layer are denoted by $\mathbf{W}_C \in \mathbb{R}^{V \times U^{(C-1)}}$, $\mathbf{b}_C \in \mathbb{R}^V$. For $2 \leq m < C$, the parameters of intermediate layer are defined by $\mathbf{W}_m \in \mathbb{R}^{U^{(m)} \times U^{(m-1)}}$, $\mathbf{b}_m \in \mathbb{R}^{U^{(m)}}$ except $\mathbf{W}_2 \in \mathbb{R}^{U^{(2)} \times V_{org}}$. By using the proposed transformation function, DRSA objective function can be denoted as follows:

$$\min_{\mathbf{B}^{(\ell)}, \boldsymbol{\theta}^{(\ell)}} \Bigg(q\Big(\mathbf{X}^{(\ell)}, \mathbf{D}^{(\ell)}, \mathbf{B}^{(\ell)}, \boldsymbol{\theta}^{(\ell)}, \boldsymbol{\Psi}^{(j)}\Big) + r\Big(\mathbf{B}^{(\ell)}\Big)\Bigg), \tag{4.7}$$

where $\boldsymbol{\Psi}^{(j)}$ is a set of randomly selected time points related to $j\text{-}th$ iteration and the size of this set ($|\boldsymbol{\Psi}^{(j)}| < T$) is equal to the batch size. The main term of DRSA method is defined as follows:

$$q\Big(\mathbf{X}, \mathbf{D}, \mathbf{B}, \boldsymbol{\theta}, \boldsymbol{\Psi}^{(j)}\Big) = \sum_{i \in \boldsymbol{\Psi}^{(j)}} \Big\| f\big(\mathbf{x}_{i.}; \boldsymbol{\theta}\big) - \mathbf{d}_{i.}\mathbf{B} \Big\|_2^2, \tag{4.8}$$

where $\mathbf{x}_{i.} \in \mathbb{R}^{1 \times V}$ denotes all voxels belong to $i\text{-}th$ time point (row) of the neural activities $\mathbf{X}$, and $\mathbf{d}_{i.} \in \mathbb{R}^{1 \times P}$ is the $i\text{-}th$ row of the design matrix $\mathbf{D}$. Unlike other applications of deep transformation function [13, 41, 42], we consider a fixed structure of deep network layers for all subjects, including $f(\mathbf{x}^{(\ell)}, \boldsymbol{\theta}^{(\ell)})$ rather than $f_\ell(\mathbf{x}^{(\ell)}, \boldsymbol{\theta}^{(\ell)})$. This notation can improve the stability of generated results and also decrease the number of parameters that must be estimated for each RSA problem. Thus, we just need to estimate additional network parameters ($\boldsymbol{\theta}^{(\ell)}, 1 \leq \ell \leq S$) for each problem in comparison with the classical RSA approaches. Further, DRSA regularization term is defined as follows, where $\alpha \geq 1$ is the scaling factor that must be defined based on data normalization:

$$r\Big(\mathbf{B}\Big) = \sum_{j=1}^{V} \sum_{k=1}^{P} \alpha \Big|\beta_{kj}\Big| + 10\alpha\Big(\beta_{kj}\Big)^2. \tag{4.9}$$

Here, we propose an optimization approach for DRSA problem. This method seeks an optimum solution for (4.7) by using two different steps, which iteratively work in unison. By considering fixed network parameters ($\boldsymbol{\theta}^{(\ell)}$), RSA objective function (J_R) is firstly optimized as follows by using the $j\text{-}th$ mini-batch ($\boldsymbol{\Psi}^{(j)}$) of neural activities:

$$\min_{\mathbf{B}^{(\ell)}} \Bigg(J_R^{(\ell)} = q\Big(\mathbf{X}^{(\ell)}, \mathbf{D}^{(\ell)}, \mathbf{B}^{(\ell)}, \boldsymbol{\theta}^{(\ell)}, \boldsymbol{\Psi}^{(j)}\Big) + r\Big(\mathbf{B}^{(\ell)}\Big)\Bigg). \tag{4.10}$$

Lemma 4.1. *For minimizing J_R, we have:*

$$\nabla_{\mathbf{B}}\Big(J_R\Big) = \alpha\,\mathrm{sign}\,\Big(\mathbf{B}\Big) + 20\alpha\mathbf{B} - 2\sum_{i\in\boldsymbol{\Psi}^{(j)}} \mathbf{d}_{i.}^{\top}\Big(f\big(\mathbf{x}_{i.};\boldsymbol{\theta}\big) - \mathbf{d}_{i.}\mathbf{B}\Big) \tag{4.11}$$

where $\mathrm{sign}\,\Big(\mathbf{B}\Big) \in \Big\{-1,+1\Big\}^{P\times V}$ *is the sign function.*

Proof.

$$\nabla_{\mathbf{B}}\Big(J_R\Big) = \frac{\partial}{\partial\mathbf{B}}q\Big(\mathbf{X},\mathbf{D},\mathbf{B},\boldsymbol{\theta},\boldsymbol{\Psi}^{(j)}\Big) + \frac{\partial}{\partial\mathbf{B}}r\Big(\mathbf{B}\Big)$$

So, we have:

$$\frac{\partial}{\partial\mathbf{B}}q\Big(\mathbf{X},\mathbf{D},\mathbf{B},\boldsymbol{\theta},\boldsymbol{\Psi}^{(j)}\Big) = \sum_{i\in\boldsymbol{\Psi}^{(j)}}\frac{\partial}{\partial\mathbf{B}}\Big\|f\big(\mathbf{x}_{i.};\boldsymbol{\theta}\big) - \mathbf{d}_{i.}\mathbf{B}\Big\|_2^2 =$$

$$2\sum_{i\in\boldsymbol{\Psi}^{(j)}}\frac{\partial}{\partial\mathbf{B}}\Bigg(f\big(\mathbf{x}_{i.};\boldsymbol{\theta}\big) - \mathbf{d}_{i.}\mathbf{B}\Bigg)^{\top}\Bigg(f\big(\mathbf{x}_{i.};\boldsymbol{\theta}\big) - \mathbf{d}_{i.}\mathbf{B}\Bigg) =$$

$$2\sum_{i\in\boldsymbol{\Psi}^{(j)}}\Bigg(\frac{\partial}{\partial\mathbf{B}}f\big(\mathbf{x}_{i.};\boldsymbol{\theta}\big) - \frac{\partial}{\partial\mathbf{B}}\mathbf{d}_{i.}\mathbf{B}\Bigg)^{\top}\Bigg(f\big(\mathbf{x}_{i.};\boldsymbol{\theta}\big) - \mathbf{d}_{i.}\mathbf{B}\Bigg),$$

where $\frac{\partial}{\partial\mathbf{B}}f\big(\mathbf{x}_{i.};\boldsymbol{\theta}\big) = 0$ and $\frac{\partial}{\partial\mathbf{B}}\mathbf{d}_{i.}\mathbf{B} = \mathbf{d}_{i.}$, so we have:

$$\frac{\partial}{\partial\mathbf{B}}q\Big(\mathbf{X},\mathbf{D},\mathbf{B},\boldsymbol{\theta},\boldsymbol{\Psi}^{(j)}\Big) = -2\sum_{i\in\boldsymbol{\Psi}^{(j)}}\mathbf{d}_{i.}^{\top}\Bigg(f\big(\mathbf{x}_{i.};\boldsymbol{\theta}\big) - \mathbf{d}_{i.}\mathbf{B}\Bigg)$$

As the next term, we have to calculate the derivation of the regularization function:

$$\frac{\partial}{\partial\mathbf{B}}r\Big(\mathbf{B}\Big) = \alpha\sum_{j=1}^{V}\sum_{k=1}^{P}\frac{\partial}{\partial\mathbf{B}}\Big|\beta_{kj}\Big| + 10\alpha\sum_{j=1}^{V}\sum_{k=1}^{P}\frac{\partial}{\partial\mathbf{B}}\beta_{kj}^2$$

By considering the following term:

$$\frac{\partial}{\partial\Big(\mathbf{B}\Big)_{i\ell}}\Big|\beta_{kj}\Big| = \begin{cases} {}^{|\beta_{kj}|}/_{\beta_{kj}} = \mathrm{sign}(\beta_{kj}) & {\scriptstyle k=i,\,j=\ell} \\ 0 & \text{otherwise} \end{cases},$$

where we assume $\mathrm{sign}(0) = 1$ and have:

$$\sum_{j=1}^{V}\sum_{k=1}^{P}\frac{\partial}{\partial\mathbf{B}}\Big|\beta_{kj}\Big| = \mathrm{sign}\,\big(\mathbf{B}\big)$$

Further, we can also define following term:

$$\frac{\partial}{\partial\big(\mathbf{B}\big)_{i\ell}}\beta_{kj}^2 = \begin{cases} 2\beta_{kj} & {\scriptstyle k=i,\, j=\ell} \\ 0 & \text{otherwise} \end{cases},$$

so we have:

$$\sum_{j=1}^{V}\sum_{k=1}^{P}\frac{\partial}{\partial\mathbf{B}}\beta_{kj}^2 = 2\mathbf{B}.$$

Further, the derivation of the regularization function is

$$\frac{\partial}{\partial\mathbf{B}}r\Big(\mathbf{B}\Big) = \alpha\,\text{sign}\,\Big(\mathbf{B}\Big) + 20\alpha\mathbf{B}.$$

Thus, the whole of procedure results:

$$\nabla_{\mathbf{B}}\Big(J_R\Big) = \frac{\partial}{\partial\mathbf{B}}q\Big(\mathbf{X},\mathbf{D},\mathbf{B},\boldsymbol{\theta},\boldsymbol{\Psi}^{(j)}\Big) + \frac{\partial}{\partial\mathbf{B}}r\Big(\mathbf{B}\Big) =$$
$$\alpha\,\text{sign}\,\Big(\mathbf{B}\Big) + 20\alpha\mathbf{B} - 2\sum_{i\in\boldsymbol{\Psi}^{(j)}}\mathbf{d}_{i.}^{\top}\Big(f\big(\mathbf{x}_{i.};\boldsymbol{\theta}\big) - \mathbf{d}_{i.}\mathbf{B}\Big)$$

□

As the next step, back-propagation algorithm [56] is applied to the kernel objective function (J_K) in order to update the network parameters in the $j\text{-}th$ iteration:

$$\min_{\boldsymbol{\theta}^{(\ell)}}\left(J_K^{(\ell)} = q\Big(\mathbf{X}^{(\ell)},\mathbf{D}^{(\ell)},\mathbf{B}^{(\ell)},\boldsymbol{\theta}^{(\ell)},\boldsymbol{\Psi}^{(j)}\Big)\right), \tag{4.12}$$

where we consider the regressors matrix ($\mathbf{B}^{(\ell)}$) is fixed in this step.

Lemma 4.2. *The network parameters ($\boldsymbol{\theta}$) can be updated by considering the vector $\mathbf{d}_{i.}\mathbf{B}$ as the ground truth of $f\big(\mathbf{x}_{i.};\boldsymbol{\theta}\big)$ and then using back-propagation algorithm to update the parameters.*

Proof. Since $f\big(\mathbf{x}_{i.};\boldsymbol{\theta}\big)$ is a standard multilayer perceptron (MLP), we can update the network parameters by using $\nabla_{\boldsymbol{\theta}}\Big(f\big(\mathbf{x}_{i.};\boldsymbol{\theta}\big)\Big)$, where the output of the optimized deep neural network has the lowest error in comparison with the vector $\mathbf{d}_{i.}\mathbf{B}$ (as the ground truth). By reducing this error, J_K will be also minimized. Please refer [56] for technical information related to MLP and back-propagation algorithm. □

Algorithm 4.8 Deep RSA for $\ell - th$ subject

Input: Data $\mathbf{X}^{(\ell)}$, Design $\mathbf{D}^{(\ell)}$, Number of layers C, Number of units $U^{(m)}$ for $m = 2{:}C$, Learning rate η (default 10^{-3}), Maximum Iteration M (default 1000), Batch Size N (default 50), Scaling parameter α (default 10), Adam optimization parameters $\mu_1 = 0.9$, $\mu_2 = 0.999$, $\epsilon = 10^{-8}$ [106].

Output: Regressors matrix $\mathbf{B}^{(\ell)}$, and Parameters $\boldsymbol{\theta}^{(\ell)}$

Method:

```
Initialize $\boldsymbol{\theta}^{(\ell)} \sim \mathcal{N}(0,1)$, $\mathbf{B}^{(\ell)} \sim \mathcal{N}(0,1)$.
$\delta_0 \leftarrow 0$ (Initialize $1^{st}$ moment vector)
$\gamma_0 \leftarrow 0$ (Initialize $2^{nd}$ moment vector)
FOR $j = 1{:}M$
    Create $\boldsymbol{\Psi}^{(j)}$ by selecting $N$ samples from 1 to $T$.
    $\widehat{\phi}_j = \sum_{i \in \boldsymbol{\Psi}^{(j)}} \nabla_{\mathbf{B}^{(\ell)}} \Big( J_R^{(\ell)} \Big)$.
    Update $\mathbf{B}^{(\ell)} \leftarrow \mathbf{B}^{(\ell)} - \eta \widehat{\phi}_j$.
    $\phi_j = \sum_{i \in \boldsymbol{\Psi}^{(j)}} \nabla_{\boldsymbol{\theta}^{(\ell)}} \Big( f \big( \mathbf{x}_{i.}^{(\ell)}; \boldsymbol{\theta}^{(\ell)} \big) \Big)$.
    $\delta_j \leftarrow \mu_1 \delta_{j-1} + (1 - \mu_1) \phi_j$.
    $\gamma_j \leftarrow \mu_2 \gamma_{j-1} + (1 - \mu_2) \phi_j^2$.
    $\widetilde{\delta}_j \leftarrow {}^{\delta_j}\!/_{(1 - \mu_1^j)}$.
    $\widetilde{\gamma}_j \leftarrow {}^{\gamma_j}\!/_{(1 - \mu_2^j)}$.
    Update $\boldsymbol{\theta}^{(\ell)} \leftarrow \boldsymbol{\theta}^{(\ell)} - \eta \widetilde{\delta}_j \Big/ (\sqrt{\widetilde{\gamma}_j} - \epsilon)$.
END FOR.
```

Algorithm (4.8) illustrates the whole of the optimization procedure. As the first step, the network parameters are considered fixed, and then Stochastic Gradient Descent (SGD) [13, 42, 107] updates the regressors matrix $\mathbf{B}^{(\ell)}$ belong to ℓ-th subject. As the second step, the regressors matrix is assumed fixed, and then the Adam [106] approach updates the deep network parameters. In order to generalize DRSA for the multi-subject problem, we also employ (4.2) for calculating the mean of the regressors matrices across subjects.

In summary, this section develops DRSA as a flexible deep approach for improving the performance of representational similarity analysis (RSA) method in fMRI analysis. For seeking an efficient analysis, DRSA uses a deep network (*multiple stacked layers of nonlinear transformation*) for mapping neural activities of each subject into an embedded space ($f : \mathbb{R}^{V_{org}} \rightarrow \mathbb{R}^{V}$). Unlike the

Table 4.1 Mean of square error (MSE) across subject, standard deviation of MSE for all methods is lower than 10^{-2}

Datasets	RSA	LASSO	Elastic Net	OWL	BRSA	GRSA	DRSA
R105	0.984	0.874	0.864	0.812	0.785	0.701	**0.452**
R107	0.971	0.868	0.831	0.789	0.832	0.752	**0.632**
R232	0.999	1.000	0.990	0.895	0.764	0.652	**0.435**
W001	0.985	0.953	0.965	0.916	0.893	0.831	**0.691**
W002D	0.953	0.632	0.722	0.483	0.677	0.538	**0.156**
W002P	0.948	0.732	0.738	0.708	0.509	0.235	**0.137**
W005	0.989	0.899	0.864	0.831	0.863	0.746	**0.372**
W011D	0.973	0.921	0.902	0.874	0.821	0.802	**0.582**
W011W	0.968	0.898	0.834	0.784	0.800	0.712	**0.699**
W017	0.988	0.602	0.681	0.737	0.599	0.430	**0.288**
W052R	0.996	0.915	0.745	0.482	0.721	0.143	**0.066**
W052W	0.997	0.810	0.777	0.503	0.671	0.204	**0.050**
W102	0.989	0.771	0.691	0.231	0.372	0.389	**0.160**
W105	0.991	0.914	0.876	0.843	0.798	0.734	**0.324**
W107	0.973	0.972	0.952	0.900	0.892	0.712	**0.468**
W116A	0.990	0.918	0.943	0.897	0.851	0.832	**0.783**
W116V	0.988	0.950	0.942	0.895	0.893	0.800	**0.431**
W164	0.970	0.688	0.614	0.532	0.395	0.375	**0.286**
W231	0.986	0.853	0.832	0.773	0.731	0.733	**0.594**
W232	0.381	0.345	0.340	0.302	0.289	0.277	**0.195**

previous nonlinear methods that used a restricted fixed transformation function, mapping functions in DRSA are flexible across subjects because they employ multi-layer neural networks, which can implement any nonlinear function [13, 41, 42]. Therefore, DRSA does not suffer from disadvantages of the previous nonlinear approach. Finally, DRSA can handle a large number of subjects by using the proposed optimization algorithm, including gradient-based optimization approaches.

4.3 Experiments

The empirical studies are presented in this section. For generating the experiments, we employ the original RSA by using GLM method as a baseline. Further, we report the performance of LASSO

Table 4.2 Maximum of between-class covariance (CV) across subjects (max±std)

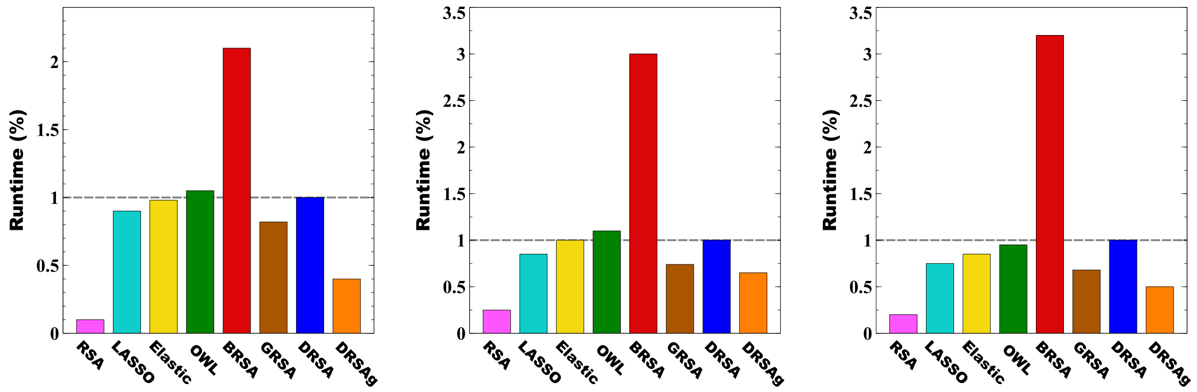
Datasets	RSA	LASSO	Elastic Net	OWL	BRSA	GRSA	DRSA
R105	2.732±0.123	0.523±0.139	0.532±0.140	0.426±0.076	0.598±0.132	0.276±0.082	**0.140±0.018**
R107	1.629±0.113	0.357±0.078	0.355±0.078	0.242±0.082	0.289±0.072	0.173±0.081	**0.096±0.011**
R232	0.030±0.003	0.027±0.001	0.026±0.001	**0.024±0.012**	0.030±0.023	0.028±0.032	0.026±0.023
W001	3.831±1.042	0.245±0.104	0.249±0.105	0.091±0.072	0.062±0.020	0.092±0.032	**0.013±0.010**
W002D	2.124±0.042	0.889±0.053	0.801±0.081	0.751±0.023	0.843±0.017	0.563±0.099	**0.102±0.002**
W002P	2.941±0.014	0.937±0.021	0.838±0.012	0.800±0.067	0.261±0.043	0.489±0.055	**0.090±0.009**
W005	0.262±0.018	0.226±0.013	0.240±0.027	0.200±0.029	0.182±0.052	0.107±0.028	**0.087±0.003**
W011D	0.471±0.073	0.162±0.027	0.162±0.023	0.142±0.062	0.152±0.023	0.051±0.043	**0.019±0.032**
W011W	0.766±0.124	0.255±0.023	0.258±0.031	0.232±0.043	0.127±0.012	0.102±0.021	**0.027±0.013**
W017	0.493±0.031	0.194±0.077	0.237±0.093	0.179±0.071	0.351±0.041	0.230±0.071	**0.053±0.010**
W052R	1.334±0.092	0.377±0.094	0.357±0.011	0.246±0.031	0.144±0.006	0.199±0.020	**0.001±0.001**
W052W	1.774±0.087	0.419±0.026	0.396±0.072	0.305±0.041	0.302±0.061	0.108±0.026	**0.011±0.007**
W102	2.111±0.048	0.235±0.000	0.200±0.004	0.208±0.023	0.193±0.054	0.242±0.054	**0.019±0.002**
W105	4.788±0.592	0.216±0.093	0.217±0.032	0.159±0.011	0.195±0.032	0.099±0.096	**0.025±0.046**
W107	1.839±0.125	0.362±0.049	0.331±0.044	0.172±0.021	0.232±0.074	0.251±0.051	**0.042±0.006**
W116A	0.505±0.106	0.147±0.003	0.143±0.020	0.120±0.039	0.059±0.001	0.102±0.048	**0.018±0.000**
W116V	2.224±0.864	0.021±0.012	0.021±0.005	0.021±0.003	0.054±0.002	0.020±0.010	**0.019±0.011**
W164	8.890±0.087	0.174±0.004	0.169±0.031	0.108±0.009	0.289±0.072	0.100±0.006	**0.037±0.004**
W231	0.438±0.072	0.325±0.052	0.325±0.023	0.300±0.037	0.126±0.045	0.272±0.076	**0.014±0.013**
W232	0.016±0.003	0.014±0.020	**0.010±0.007**	0.023±0.002	0.038±0.021	0.034±0.025	0.030±0.020

algorithm [31], where parameter $\alpha = 0.9$ generates the best results. As another regularized method, Elastic Net is also used for evaluating the proposed method. In this method, the best results are achieved by $\alpha = 1.0$ and $\rho = 0.5$. As the last method with regularization term, the performance of OWL is also presented. To generate the results, *Spike* weight sequence is employed for OWL method, as the best approach in the original paper [34]. As one of the Bayesian approaches, the performance of Bayesian RSA (BRSA) is also analyzed. All parameters in this method are assigned optimum based on the original paper [3]. Moreover, the number of iterations for all of the mentioned methods is considered 2000. This thesis reports the performance of the proposed method with two different transformation function, including linear, and deep. As the linear approach, Gradient RSA (GRSA) utilizes the objective function (4.7) and optimization Algorithm (4.8), but the transformation function is considered linear, i.e., $f\left(\mathbf{x}\right) = \mathbf{x}$. The aim of reporting GRSA is illustrating how much the proposed

Table 4.3 Maximum of between-class correlation (CR) across subjects (max±std)

Datasets	RSA	LASSO	Elastic Net	OWL	BRSA	GRSA	DRSA
R105	1.147±0.042	0.751±0.242	0.731±0.212	0.821±0.120	0.389±0.010	0.451±0.081	**0.372±0.016**
R107	0.922±0.053	0.715±0.147	0.718±0.148	0.531±0.123	0.458±0.076	0.142±0.092	**0.135±0.000**
R232	1.027±0.033	0.900±0.026	0.876±0.210	0.631±0.193	0.871±0.100	0.555±0.112	**0.496±0.093**
W001	0.767±0.119	**0.289±0.126**	0.292±0.147	0.302±0.021	0.584±0.043	0.324±0.041	0.307±0.010
W002D	2.124±0.045	0.389±0.071	0.411±0.054	0.289±0.025	0.319±0.011	0.432±0.025	**0.102±0.005**
W002P	2.941±0.011	0.481±0.067	0.481±0.032	0.361±0.004	0.500±0.000	0.099±0.006	**0.090±0.012**
W005	0.891±0.035	0.823±0.036	0.739±0.036	0.699±0.076	0.519±0.045	0.361±0.021	**0.139±0.019**
W011D	0.887±0.093	0.690±0.131	0.621±0.092	0.588±0.010	0.610±0.037	0.593±0.027	**0.165±0.076**
W011W	0.827±0.032	0.607±0.058	0.507±0.021	0.436±0.071	0.591±0.068	0.367±0.100	**0.253±0.051**
W017	0.493±0.062	0.365±0.032	0.341±0.026	0.420±0.008	0.075±0.003	0.105±0.041	**0.053±0.019**
W052R	1.334±0.039	0.377±0.069	0.384±0.021	0.209±0.037	0.106±0.052	0.231±0.075	**0.015±0.000**
W052W	1.774±0.099	0.692±0.041	0.471±0.052	0.296±0.019	0.111±0.003	0.182±0.041	**0.011±0.001**
W102	2.117±0.072	0.355±0.088	0.404±0.051	0.352±0.006	0.080±0.009	0.266±0.061	**0.019±0.005**
W105	1.158±0.074	0.696±0.119	0.621±0.099	0.569±0.073	0.724±0.069	0.639±0.031	**0.461±0.112**
W107	0.917±0.031	0.652±0.074	0.520±0.008	0.366±0.029	0.444±0.091	0.359±0.055	**0.160±0.021**
W116A	0.951±0.045	0.851±0.077	0.799±0.038	0.572±0.155	0.383±0.051	0.451±0.012	**0.270±0.039**
W116V	0.637±0.105	0.472±0.163	0.459±0.191	0.327±0.076	0.231±0.021	**0.132±0.021**	0.328±0.011
W164	8.890±0.108	0.241±0.057	0.228±0.081	0.342±0.002	0.176±0.021	0.174±0.004	**0.037±0.002**
W231	0.888±0.113	0.761±0.114	0.700±0.172	0.421±0.273	0.521±0.082	0.231±0.063	**0.126±0.002**
W232	1.024±0.046	0.902±0.028	0.893 ±0.021	0.711±0.121	0.421±0.121	0.621±0.094	**0.235±0.016**

method can improve the performance of RSA analysis without deep transformation. Finally, we have presented the performance of DRSA. We generate results in both GRSA and DRSA by using different values of $\alpha = [1, 5, 10, 20, 50, 100]$. In all datasets with normalization $\mathbf{X}^{(\ell)} \sim \mathcal{N}(0, 1)$, $\alpha = 10$ has generated better trade-off between covariance and correlation in comparison with other values. Moreover, we evaluate the performance of DRSA by employing different nonlinear activation functions, including ReLU, sigmoid, $tanh$. In most of the normalized datasets, the proposed method by using sigmoid activation function has generated better performance. In GRSA and DRSA, the number of iterations is considered 1000, the batch size is assigned 50, learning rates is 10^{-3} for normalized datasets, and the Adam optimization parameters are set optimum based on the original paper [106], including $\mu_1 = 0.9$, $\mu_2 = 0.999$, and $\epsilon = 10^{-8}$. In this section, all algorithms are implemented by



(a) R105 (b) R107 (c) R232

Figure 4.2 Runtime Analysis

Python 3 on a PC with certain specifications[1] by authors for generating the experimental results.

4.3.1 Performance analysis

Table 4.1 shows the benchmarking of different RSA algorithms by using mean of square error (MSE), i.e., metric (4.3). In this table, the standard deviation of all RSA methods is lower than 10^{-2}. Table 4.2 has also analyzed the maximum of between-class covariance by using (4.5). Further, Table 4.3 has evaluated the maximum of between-class correlation by utilizing (4.4). In this section, we have employed two hidden layers for DRSA, i.e., $C = 4$. Here, we present the number of units for the hidden layers and the output layer as follows, $[HL1, HL2, OUT]$. As an example, $[1000, 700, 500]$ represents $U^{(1)} = 1000$ as the number of the first hidden layer, $U^{(2)} = 700$ as the number of the second hidden layer and $V = 500$ as the output layer. The best results for $R107$ are achieved by $[400, 200, 100]$. While we can use any format for the structure of the deep network, the rest of DRSA experiments are generated by using $[1000, 700, 500]$ setting. In practice, this structure can provide an efficient trade-off between runtime and performance for DRSA method. As depicted in the result tables, DRSA has generated better performance in comparison with other methods because it firstly provides better feature representation in the linear embedded space and then effectively estimates the similarities between different cognitive tasks (please also compare the performance of GRSA with other RSA methods).

[1] CPU = Xeon E5-2630, RAM = 64GB, GPU = GeForce TITAN X, OS = KDE Neon 16.04.3, CUDA = 9.0, CuDNN = 7.0.5, Python = 3.6.5, Pip = 9.0.3, Numpy = 1.14.2, Scipy = 1.0.1, Scikit-Learn = 0.19.1, Tensorflow = 1.7.0.

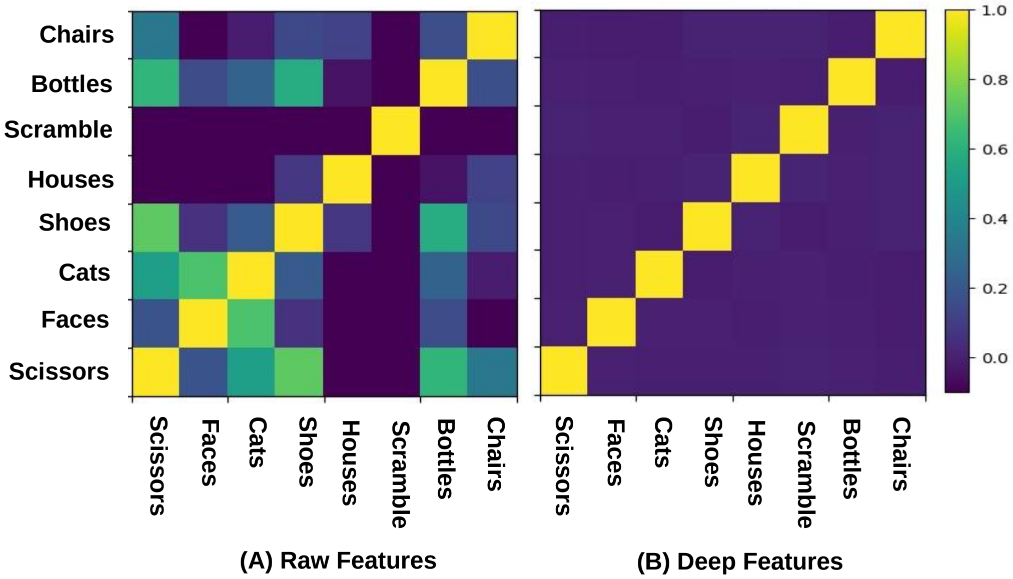
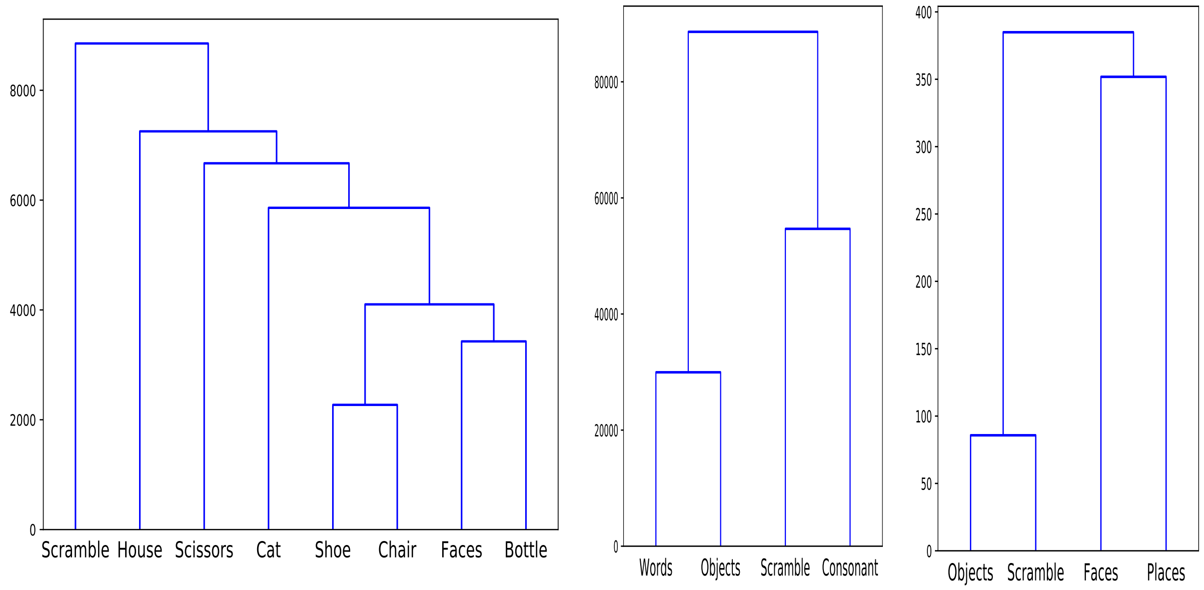


Figure 4.3 Comparing correlation of raw and deep features by using R105

4.3.2 Runtime analysis

This section analyzes the runtime of the proposed method and other RSA methods by employing ROI-based datasets. As mentioned before, all results in this section are generated by using a PC with certain specifications. Further, the runtime of DRSA is evaluated by using both hardware, i.e., CPU (DRSA) and GPU (DRSAg). Figure 4.2 demonstrates the runtime of the mentioned methods, where runtime of other algorithms are scaled based on DRSA. In other words, the runtime of the proposed method is considered as a unit. As illustrated in this figure, BRSA generated the worse runtime because it must estimate a wide range of hyper-parameters for high-dimensional datasets. Further, the runtime of DRSA is similar to the regularized methods (LASSO, Elastic Net, OWL), while those algorithms did not utilize any transformation function. Since GRSA (same as DRSA) employs a min-batch of time-points, it produces better runtime in comparison with the regularized methods. By considering the performance of the proposed method in the previous section, DRSA generates acceptable runtime. As mentioned before, the proposed method utilizes gradient-based approaches that can rapidly reduce the time complexity of the optimization procedure. It is worth noting that runtime of whole-brain datasets has the same tendency.



(A) R105

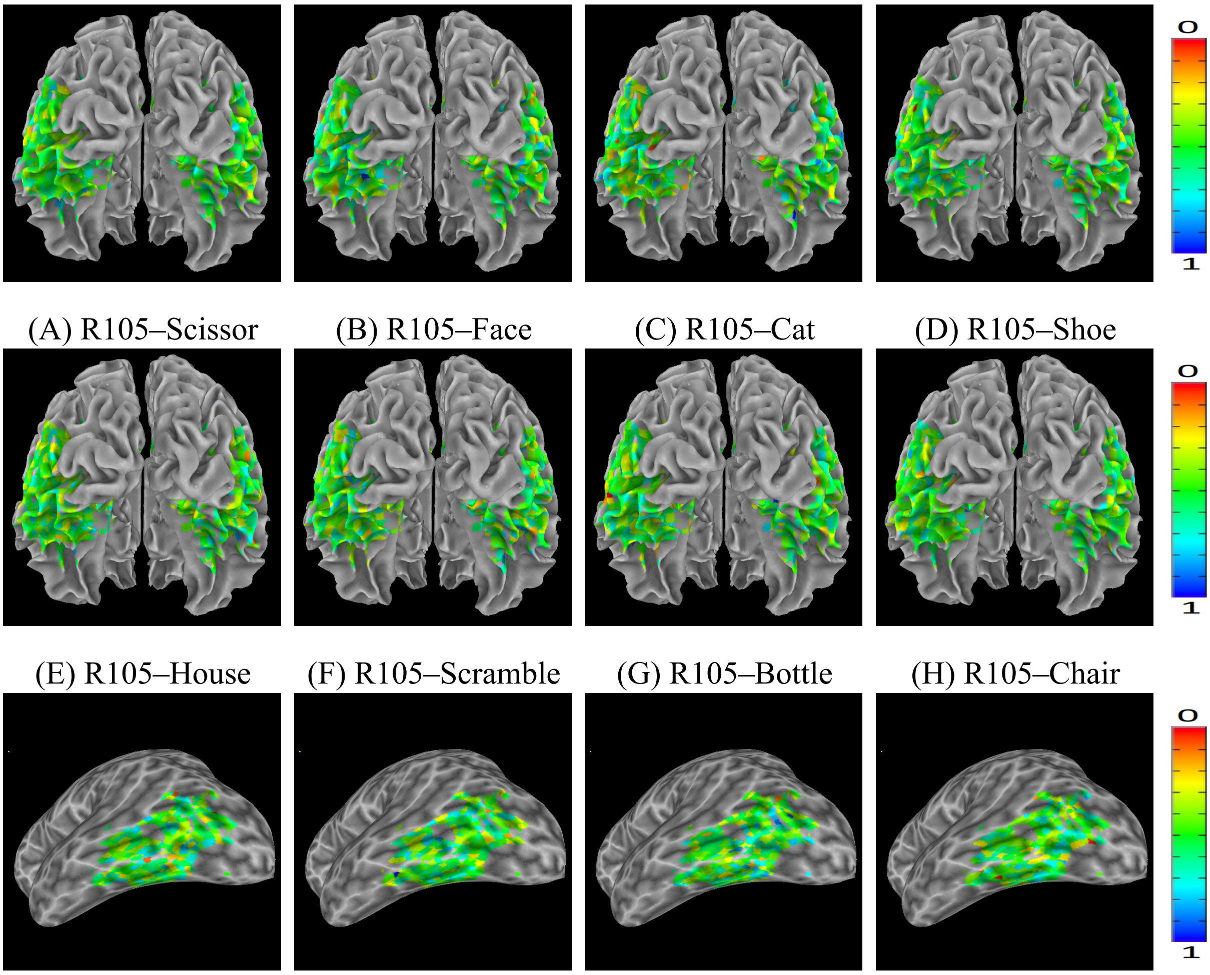
(B) R107

(C) R232

Figure 4.4 Supervised similarity analysis

4.3.3 Visualizing Supervised Similarity

In this section, we show why DRSA can provide better similarity analysis in comparison with other linear methods. Based on R105 data, Figure 4.3 illustrates the correlation matrices in both the original space and the embedded space that is generated by using the deep kernel. As this figure depicted, the cell located in i-th row and j-th column shows how much their corresponding categories of stimuli are dependent on each other. As mentioned before, the deep kernel in DRSA maps the neural activities to an embedded space, where the similarity analysis can be applied by using a linear clustering (model) approach. Further, this figure demonstrates that the extracted features are linearly independent in the embedded space. Indeed, we have reported the maximum value of these cells in Table 4.3 and Table 4.2 across each of RSA techniques. Here, smaller correlation /or covariance values represent better similarity analysis. Moreover, we can show how much the neural signature in the embedded space is information-rich. Figure 4.4 visualizes the supervised similarity of the region-based datasets by using the generated $\mathbf{B}^*$ matrix in the DRSA method. As this figure illustrated, DRSA can generate different nested similarity analysis in comparison with WSCE by using supervised information.



(A) R105–Scissor (B) R105–Face (C) R105–Cat (D) R105–Shoe

(E) R105–House (F) R105–Scramble (G) R105–Bottle (H) R105–Chair

(I) R107–Word (J) R107–Object (K) R107–Consonant (L) R107–Scramble

Figure 4.5 Representing neural activities of selected snapshots in data R105

4.3.4 Representing Neural Activities

In this section, we visualize some samples of neural activities based on $\mathbf{B}^*$ matrix that is generated for datasets R105, and R107. As Figure 4.5 shows the generated neural signatures. Here, we demonstrate the left side of the brain for R107 because there is no ROI region on the right side for this dataset [108]. Like the previous chapter, while some of the neural activities are significantly distinctive across categories of stimuli, the rest of them are highly correlated.

4.3.5 Whole-brain data analysis

One advantage of using whole-brain data is comparing different datasets. Previously, the main challenge for analyzing these data was the number of dimensions, i.e., the raw-voxels are noisy and

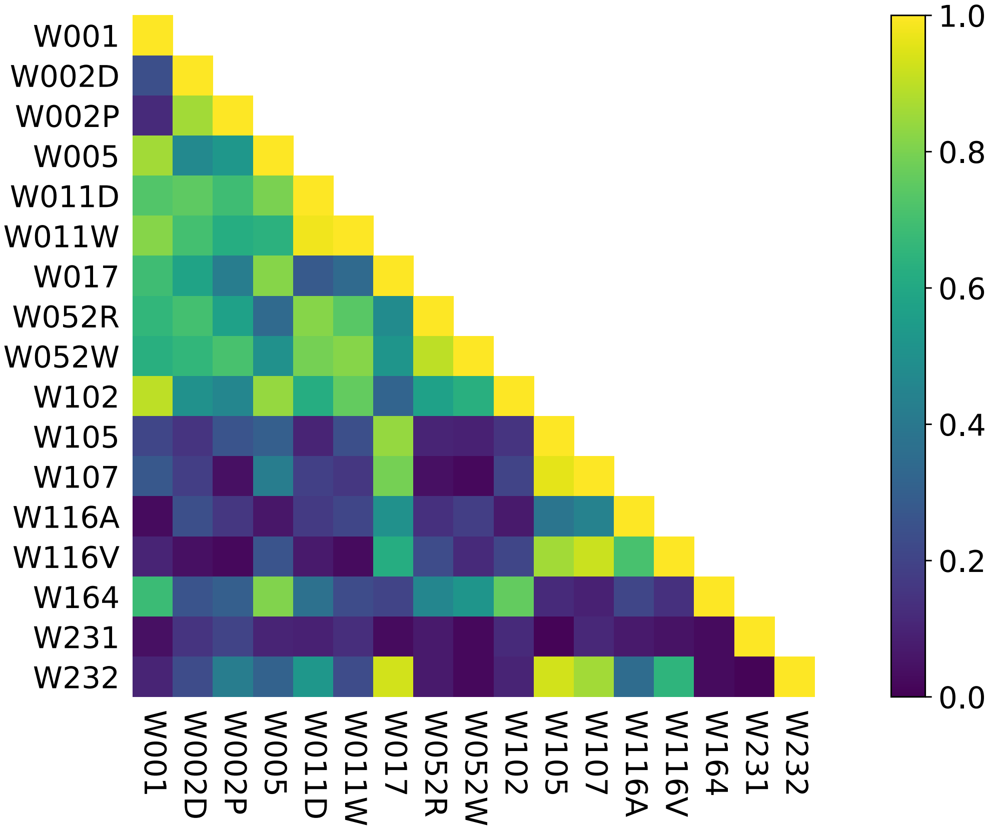


Figure 4.6 Comparing the similarities (or differences) across whole-brain datasets

sparse [3]. As the previous sections demonstrate, the classical methods cannot provide stable performance for analyzing whole-brain data. The proposed method can improve both the performance and runtime of whole-brain analysis by mapping the neural activities to an embedded linear space. As mentioned before, the number of original voxels in all of the whole-brain data is $V_{org} = 19742$. In the previous section, the deep transformation function maps these voxels to a linear embedded space with $V = 500$ dimensions by utilizing a network with $[1000, 700, 500]$ structure. We believe that not only can this transformation function significantly reduce the noise and sparsity but also the extracted neural signatures are information-rich. Thus, we compare the distance of these signatures across datasets. Figure 4.6 shows this comparison, where (for each pair of datasets) the distances between different categories of the first dataset are compared with all categories of the second datasets, i.e., $\sum_{i=1}^{P_1}\sum_{j=1}^{P_2}\left\|\beta_{i.}^{(*,1)} - \beta_{j.}^{(*,2)}\right\|_2^2$, $\beta_{i.}^{(*,k)} \in \mathbf{B}^{(*,k)}$. Here, the regressors matrix $\mathbf{B}^{(*,k)}$ is related to k-th datasets and calculated by using (4.2). As Figure 4.6 shows, the datasets related to each type of cognitive tasks (visual stimuli, decision making, and flavor) have more within-category similarity.

4.4 Conclusion

This section extended a deep approach for Representational Similarity Analysis (RSA) methods in order to provide accurate similarity (or distance) analysis in multi-subject fMRI data. Deep Representational Similarity Analysis (DRSA) can handle fMRI datasets with noise, sparsity, nonlinearity, high-dimensionality (broad ROI or whole-brain data), and a large number of subjects. DRSA utilizes gradient-based optimization approaches and generates an efficient runtime on large datasets. Further, DRSA is not limited by a restricted fixed representational space because the transformation function in DRSA is a multi-layer neural network, which can separately implement any nonlinear function for each subject to transfer the neural activities to an embedded linear space. To evaluate the performance of the proposed method, multi-subject fMRI datasets with various tasks–including visual stimuli, decision making, flavor, and working memory–are employed for running the empirical studies. And, the results confirm that DRSA achieves superior performance to other state-of-the-art RSA algorithms for evaluating the similarities between distinctive cognitive tasks.

Chapter 5. Imbalance Multi-Voxel Pattern Analysis

In order to decode neural activities in the human brain, Multi-Voxel Pattern Analysis (MVPA) technique [19, 20, 85] must apply machine learning methods to task-based functional Magnetic Resonance Imaging (fMRI) datasets. In this section, we develop a modified version of imbalance Adapting Boosting (AdaBoost) algorithm for binary classification. This algorithm uses a supervised random sampling and penalty values, which are calculated by the correlation between different classes, for improving the performance of prediction. This binary classification will be used in a one-versus-all Error-Correcting Output Codes (ECOC) method as a multiclass approach for classifying the categories of the brain response.

5.1 Imbalance AdaBoost Binary Classification (IABC)

In previous sections, we mentioned the imbalance issue in the MVPA analysis. In practice, there are two approaches in order to deal with this issue, i.e. designing an imbalance classifier, or converting the imbalance problem to an ensemble of balance classification models. Previous studies demonstrated that the performance of imbalance classifiers may not be stable, especially when we have sparsity and noise in our datasets [19, 20, 109]. Since fMRI datasets mostly include noise and sparsity, this paper has chosen the ensemble approach. Technically, ensemble learning also contains two groups of solutions, i.e. bagging or boosting. While bagging (such as our method in Chapter 3) generates all classifiers at the same time and then combine all of them as the final model, the boosting gradually creates each classifier in order to improve the performance of each iteration by tracing errors of previous iterations. We just have to note that ensemble learning can be used in both balance and imbalance problems. In fact, the main difference comes from the strategy of sampling. In balance problems, sampling methods are applied to the whole of datasets, whereas instances of the large class are sampled in the imbalance problems [109]. As depicted in Figure 5.1, this paper presents a new branch of AdaBoost algorithm, which is called Imbalance AdaBoost Binary Classification (IABC), in order to significantly improve the performance of the final model in fMRI analysis. In a nutshell, this algorithm firstly converts an imbalance MVPA problem to a set of balance problems. Then, it

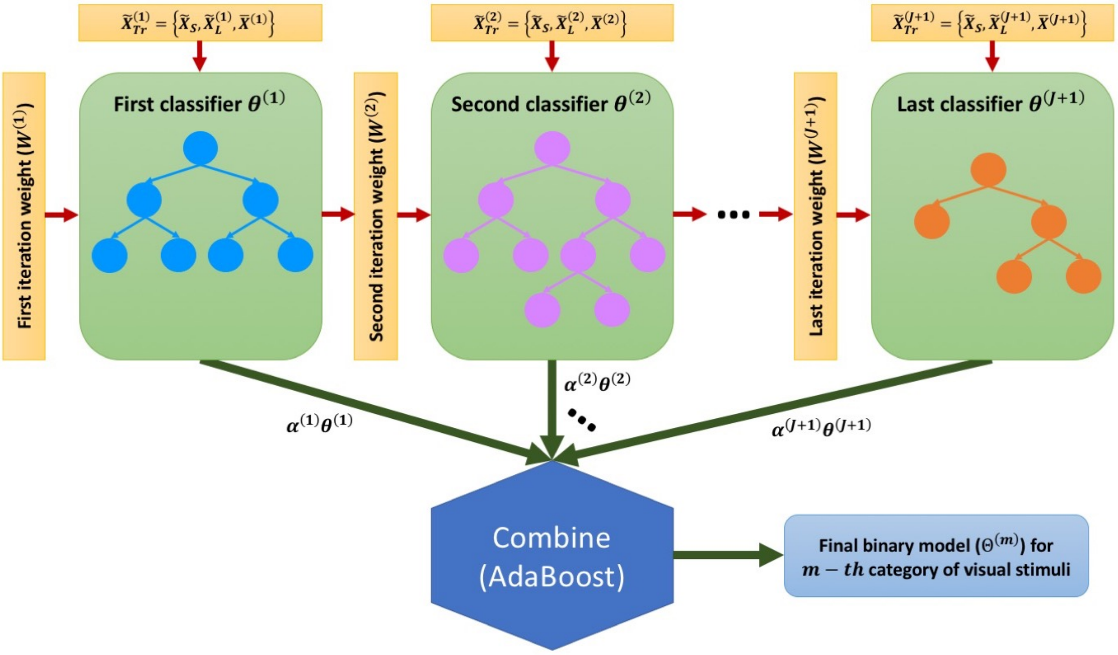

$\widetilde{X}_{Tr}^{(1)} = \{\widetilde{X}_S, \widetilde{X}_L^{(1)}, \overline{X}^{(1)}\}$
$\widetilde{X}_{Tr}^{(2)} = \{\widetilde{X}_S, \widetilde{X}_L^{(2)}, \overline{X}^{(2)}\}$
$\widetilde{X}_{Tr}^{(J+1)} = \{\widetilde{X}_S, \widetilde{X}_L^{(J+1)}, \overline{X}^{(J+1)}\}$
First iteration weight $(W^{(1)})$
First classifier $\theta^{(1)}$
Second iteration weight $(W^{(2)})$
Second classifier $\theta^{(2)}$
Last iteration weight $(W^{(J+1)})$
Last classifier $\theta^{(J+1)}$
$\alpha^{(1)}\theta^{(1)}$
$\alpha^{(2)}\theta^{(2)}$
$\alpha^{(J+1)}\theta^{(J+1)}$
Combine
(AdaBoost)
Final binary model $(\Theta^{(m)})$ for
$m-th$ category of visual stimuli

Figure 5.1 The proposed AdaBoost algorithm for applying a robust binary classification

iteratively applies the decision tree [109] to each of these balance problems. Finally, AdaBoost is used in order to generate the final model. In the proposed method, the weight of each classifier (tree) for the final combination is generated based on the error (failed predictions) of the previous iterations for gradually improving the performance of the final model.

In order to apply the binary classification, this paper randomly partitions the extracted features $\mathbf{X}$ into the training set $\widetilde{\mathbf{X}}$ and the testing set $\widehat{\mathbf{X}}$. As a new branch of AdaBoost algorithm, Algorithm 5.9 employs $\widetilde{\mathbf{X}}$ for training binary classification. Then, $\widehat{\mathbf{X}}$ is utilized for estimating the performance of the final model. As mentioned before, the binary classification for fMRI analysis is mostly imbalance, especially by using a one-versus-all strategy. Consequently, the number of samples in one of these binary classes is smaller than the other classes. As previously mentioned, this paper exploits this concept in order to solve the imbalance issue. Indeed, Algorithm 1 firstly partitions the training data $\widetilde{\mathbf{X}}$ into small $\widetilde{\mathbf{X}}_S$ and large $\widetilde{\mathbf{X}}_L$ classes (groups) based on the class labels $\mathbf{Y}^{(m)} \in \big\{+1, -1\big\}$. Here, all labels are -1 except the label of instances belong to $m\text{-}th$ category of visual stimuli. Then, it calculates the scale J of existed elements between two classes. We have to note that int() defines the floor function. As the next step, the large class is randomly partitioned into J parts. Indeed, J is the number of balance subsets generated from the imbalance dataset. Consequently, the number of the ensemble iteration is J. In each balance subset, training data $\widetilde{\mathbf{X}}_{Tr}^{(n)}$ is generated by all instances

Algorithm 5.9 Imbalance AdaBoost Binary Classification (IABC)

```
**Input:** Training set $\widetilde{\mathbf{X}}$, Class labels $\mathbf{Y}^{(m)}$.
**Output:** Set of classifiers $\Theta^{(m)}$.
**Method:**
Based on $\mathbf{Y}^{(m)}$, partitioning $\widetilde{\mathbf{X}} = \{\widetilde{\mathbf{X}}_S, \widetilde{\mathbf{X}}_L\}$
Calculating $J = \text{int}(\frac{|\widetilde{\mathbf{X}}_L|}{|\widetilde{\mathbf{X}}_S|})$.
Random sampling: $\widetilde{\mathbf{X}}_L = \big\{\widetilde{\mathbf{X}}_L^{(1)}, \widetilde{\mathbf{X}}_L^{(2)}, \dots, \widetilde{\mathbf{X}}_L^{(n)}, \dots, \widetilde{\mathbf{X}}_L^{(J)}\big\}$.
Initiate $\overline{\mathbf{X}}^{(1)} = \overline{\mathbf{Y}}^{(1)} = \emptyset$.
**For** $(n = 1 \dots J)$:
    $\widetilde{\mathbf{X}}_{Tr}^{(n)} = \big\{\widetilde{\mathbf{X}}_S, \widetilde{\mathbf{X}}_L^{(n)}, \overline{\mathbf{X}}^{(n)}\big\}$ as training-set for this iteration.
    $\widetilde{\mathbf{Y}}_{Tr}^{(n)} = \big\{\widetilde{\mathbf{Y}}_S, \widetilde{\mathbf{Y}}_L^{(n)}, \overline{\mathbf{Y}}^{(n)}\big\}$ as class labels for this iteration.
    $\mathbf{W}^{(n)} = \begin{cases} 1 & \text{for instances of } \widetilde{\mathbf{X}}_S \text{ or } \overline{\mathbf{X}}^{(n)} \\ 1 - \big|\text{corr}\big(\widetilde{\mathbf{X}}_S, \widetilde{\mathbf{X}}_L^{(n)}\big)\big| & \text{for instances of } \widetilde{\mathbf{X}}_L^{(n)} \end{cases}$
    $\theta^{(n)} = \text{classifier}\big(\widetilde{\mathbf{X}}_{Tr}^{(n)}, \widetilde{\mathbf{Y}}_{Tr}^{(n)}, \mathbf{W}^{(n)}\big)$ as weighted decision tree.
    Constructing $\overline{\mathbf{X}}^{(n+1)}$ as instances cannot truly trained in $\theta^{(n)}$.
    $\epsilon^{(n)} = \frac{|\overline{\mathbf{X}}^{(n+1)}|}{|\widetilde{\mathbf{X}}_{Tr}^{(n)}|}$ as error of classification.
    $\alpha^{(n)} = \frac{1}{2}\ln\big(\frac{1-\epsilon^{(n)}}{\epsilon^{(n)}}\big)$ AdaBoost weight for the classifier $\theta^{(n)}$.
**End For**
**Return** $\Theta^{(m)}\big(x\big) = \text{sign}\Big(\sum_{n=1}^{J+1}\alpha^{(n)}\theta^{(n)}\big(x\big)\Big)$ as the final model.
```

of the small class $\widetilde{\mathbf{X}}_S$, one of the partitioned parts of the large class $\widetilde{\mathbf{X}}_L$, and the instances of the previous iteration $\overline{\mathbf{X}}^{(n)}$, which cannot truly be trained (the failed predictions). After that, training weights for the final combination ($\mathbf{W}^{(n)} \in [0, 1]$) are calculated by using the Pearson correlation (corr$(a, b) = \frac{\text{cov}(a,b)}{\sigma_a \sigma_b}$) between training instances, where larger values increase the learning sensitivity. Indeed, these weights are always maximized for the instances of the small class and the failed instances of the previous iterations. Further, the weights of the other instances are a scale of the correlation between the large class and the small class. Therefore, these weights are updated in each iteration based on the performance of previous iterations. As the last step of each iteration, the proposed method generates a classification model ($\theta^{(n)}$) and its weight ($\alpha^{(n)}$) for the final combination. While classifier() can denote any kind of weighted classification algorithm, this paper employs a simple

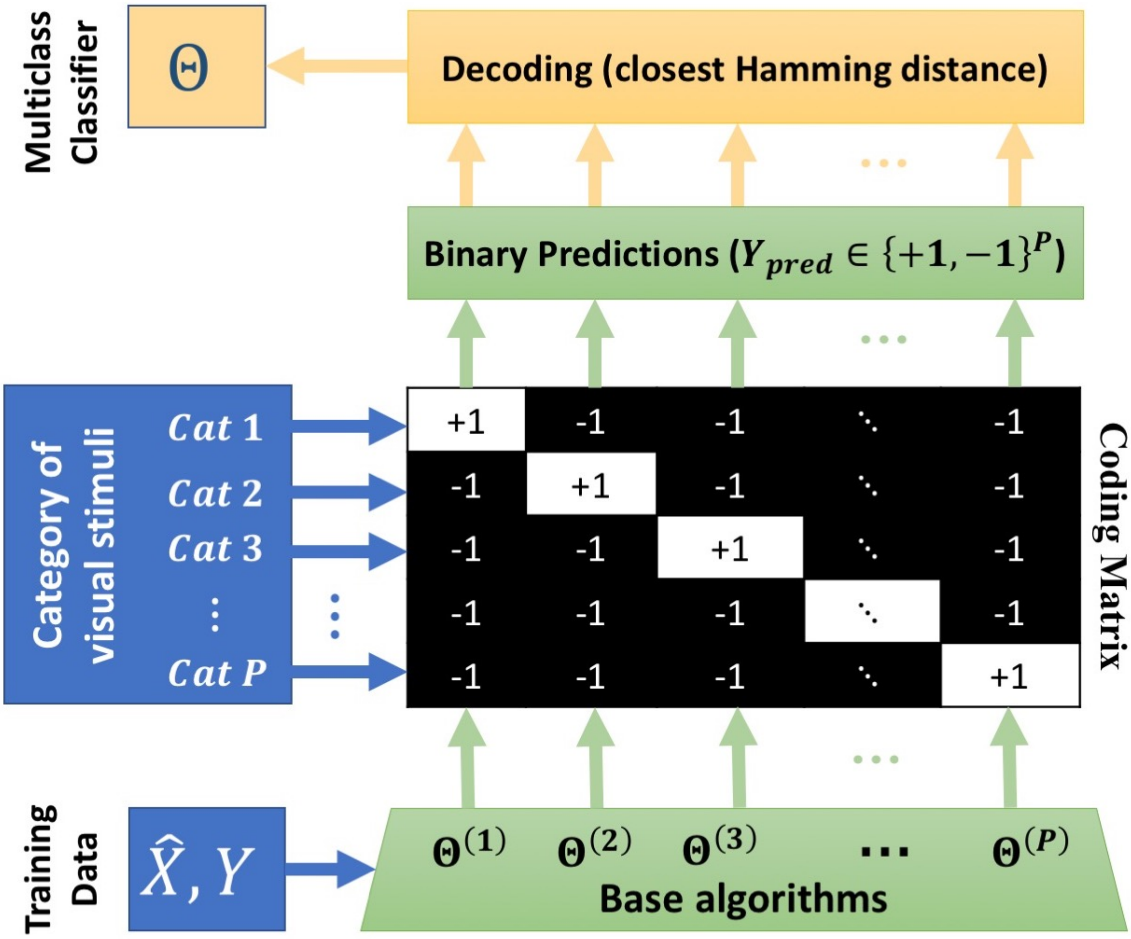

Multiclass Classifier
Θ
Decoding (closest Hamming distance)
Binary Predictions ($Y_{pred} \in \{+1,-1\}^P$)
Category of visual stimuli
Cat 1
Cat 2
Cat 3
Cat P
Coding Matrix
+1
-1
Training Data
$\hat{X}, Y$
$\Theta^{(1)}$ $\Theta^{(2)}$ $\Theta^{(3)}$ ... $\Theta^{(P)}$
Base algorithms

Figure 5.2 The proposed Error-Correcting Output Codes (ECOC) approach for multi-class classification

weighted decision tree [109] as the classification model. At the end, the final model is created by applying the AdaBoost method to the generated balance classifiers.

5.2 Multi-class IABC Classification Algorithm

In this paper, a multi-class classifier is a prediction model in order to map extracted features to the category of visual stimuli, i.e. $\Theta : \widehat{\mathbf{X}} \to \mathbf{Y}_{pred}$ where $\mathbf{Y}_{pred} \in \{1, 2, \ldots, P\}$. Generally, there are two techniques for applying multi-class classification. The first approach directly creates the classification model such as multi-class support vector machine [83] or neural network [85]. In contrast, decomposition design (indirect) uses an array of binary classifiers for solving the multi-class problems.

Based on the previous discussion related to imbalance issue in fMRI datasets, this paper utilizes Error-Correcting Output Codes (ECOC) as an indirect multi-class approach in order to extend the proposed binary classifier for the multi-class prediction. As depicted in Figure 5.2, ECOC includes three components, i.e. base algorithms, coding matrix and decoding procedures [110]. Since this paper uses one-versus-all encoding strategy, Algorithm 5.9 is employed as the based algorithms ($\Theta^{(m)}$) in the ECOC, where it generates a binary classifier for each category of visual stimuli. In other words, each independent category of the visual stimuli is compared with the rest of categories. Consequently, the size of the coding matrix is $P \times P$, where $i\text{-}th$ diagonal cell of this matrix represents the positive predictions belong to the $i\text{-}th$ category of visual stimuli and the rest of cells in this matrix determine the other categories of visual stimuli. Indeed, the number of classifiers in this strategy is exactly equal to the number of categories. As decoding stage, binary predictions, which are generated by applying the brain response to the base algorithms, are assigned to the category in the coding matrix with closest Hamming distance.

In order to present an example for ECOC procedure, consider fMRI dataset with 4 categories of visual stimuli, i.e. photos of shoes, houses, bottles, and human faces. In this problem, 4 different binary classifiers must be trained in order to distinguish each category of visual stimuli versus the rest of them (one-versus-all strategy). A 4×4 coding matrix is also generated where each diagonal element represents the positive class of these categories (classifiers). By considering the order of the coding matrix, each prediction is assigned to the closest Hamming distance in the coding matrix. In other words, if these classifiers generate the prediction $[+1, -1, -1, -1]$ for a testing instance, then this instance definitely belongs to the first category of visual stimuli. Similarly, the prediction $[-1, +1, -1, -1]$ means the instance belongs to the second category, and etc.

5.3 Experiments

5.3.1 Performance Analysis

In this section, the performance of different methods will be evaluated for both binary and multi-class analyses. In the binary analysis, the performance of the binary $\ell1$-regularized Support Vector Machine (SVM) is represented. Indeed, this method is used in [24, 83] in order to distinguish different categories of stimuli from human brain. As the regularized method that is introduced in [24] for decoding the brain patterns, the performance of the Elastic Net is also reported in this section. The parameters for Elastic Net is considered optimum based on [24]. As the method was developed in [93], the performance of a graph-based approach is also reported. Further, the performance of the

Table 5.1 Accuracy of classification analysis for evaluating different classification techniques (max±std)

Datasets	ℓ1-SVM	Elastic Net	MLP	Graph-based	IABC
R105	16.72±1.89	17.90±0.21	**40.86±0.11**	30.27±0.41	39.32±0.45
R107	26.39±2.60	26.75±0.97	37.54±0.56	34.71±0.08	**39.02±0.17**
R232	30.65±0.79	31.68±0.69	40.40±0.61	36.32±0.31	**54.59±0.02**
W001	25.47±0.36	24.42±0.14	33.73±0.84	31.02±0.67	**36.37±0.59**
W002D	61.69±0.86	62.91±0.79	64.42±0.38	64.81±0.32	**65.24±0.60**
W002P	64.98±0.15	69.55±0.61	72.59±0.80	60.85±0.70	**75.85±0.76**
W005	32.80±0.93	36.13±0.73	43.09±0.06	37.40±0.94	**49.96±0.36**
W011D	41.26±0.13	45.80±0.41	65.53±0.52	70.67±0.82	**71.78±0.09**
W011W	31.37±0.54	35.08±0.68	36.26±0.36	33.71±0.61	**38.11±0.42**
W017	22.31±0.72	28.20±0.18	40.36±0.49	43.47±0.53	**48.19±0.18**
W052R	54.83±0.17	57.19±0.52	65.45±0.66	67.02±0.07	**69.29±0.53**
W052W	53.42±0.99	55.73±0.04	67.88±0.37	64.11±0.22	**69.87±0.07**
W102	52.33±0.46	51.21±0.51	71.20±0.76	51.39±0.38	**75.36±0.42**
W105	17.72±0.10	30.18±0.39	34.12±0.41	32.06±0.17	**37.72±0.93**
W107	32.24±1.61	39.37±0.63	48.83±0.52	50.72±0.05	**54.49±0.28**
W116A	56.46±0.19	60.11±0.41	64.41±0.90	59.55±0.82	**67.93±0.34**
W116V	58.03±0.45	63.63±0.75	67.05±0.00	63.24±0.23	**70.51±0.59**
W164	50.82±0.15	53.24±0.37	63.01±0.26	57.68±0.51	**69.80±0.27**
W231	27.66±0.92	30.09±0.21	60.48±0.17	64.43±0.07	**62.62±0.38**
W232	29.97±0.46	24.88±0.88	30.35±0.26	**39.71±0.91**	30.07±0.07

proposed method is compared with Multilayer Perceptron (MLP) that was introduced [23] in order to decode the brain patterns. We have used the same network parameters that proposed in [23] as the optimized solution for MLP networks, i.e., two hidden layers with the same size of units. And, the number of units in these layers is min(T, V), where T and V are respectively the numbers of time points and voxels in the dataset. All of the mentioned algorithms are implemented in Python 3 on a PC with certain specifications[1] by authors in order to generate experimental results. Further, all evaluations are applied by using leave-one-subject-out cross-validation, except R107 that uses leave-four-subject-out strategy. As an example, we have selected brain patterns of 5 subjects in R105 for training a classifier in each iteration and then used the patterns of the rest of the subject in order to test

[1] DEL , CPU = Intel Xeon E5-2630 v3 (8×2.4 GHz), RAM = 64GB, OS = KDE Neon 16.04.4

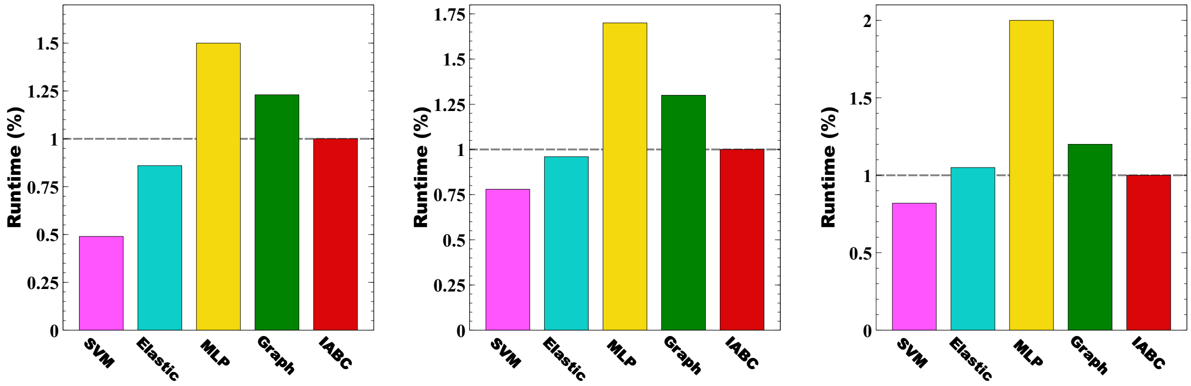
the generated cognitive model. Indeed, not only the brain patterns in training sets and testing sets are independent across subjects but also fMRI data related to each subject was separately preprocessed [89]. We have to note that the same training set and testing set are applied in each iteration to all of the evaluated methods in the whole of this thesis. Table 5.1 shows the empirical studies where Algorithm 5.9 is directly utilized for binary datasets and our multi-class approach is employed for datasets with more than two classes. As depicted in this table, the proposed method has achieved the best performance in comparison with other methods because it provided a better ensemble approach for analyzing fMRI datasets.

5.3.2 Runtime analysis

This section analyzes the runtime of the proposed method and other classification methods by employing ROI-based datasets. As mentioned before, all results in this section are generated by using a PC with certain specifications. Figure 5.3 demonstrates the runtime of the mentioned methods, where runtime of other algorithms are scaled based on IABC. In other words, the runtime of the proposed method is considered as a unit. As illustrated in this figure, MLP generated the worse runtime, specifically when the number of voxels or time points are high such as R232. Since Graph-based approach must convert the voxel space to the graph space across subjects, it cannot also generate acceptable runtime, especially for the datasets with more subjects such as R107. By considering the performance of the proposed method in the previous section, it generates suitable runtime by partitioning the huge imbalance samples to a set of small balance instances and then creating the classification (cognitive) model. It is worth noting that runtime of whole-brain datasets has the same tendency.

5.4 Conclusions

One of the primary goals in neuroscience is how the neural activities in the human brain can be mapped to the different cognitive tasks? As an interdisciplinary technique between neuroscience and computer science, Multivariate Pattern (MVP) algorithms employ task-based fMRI images for extracting and decoding brain patterns. In practice, MVP analysis can formulate as a classification problem and predict patterns of neural responses, which are generated by distinctive cognitive tasks. As the final product of an MVP analysis, decision surfaces are defined to distinguish different stimuli



(a) R105 (b) R107 (c) R203

Figure 5.3 Runtime Analysis

in the human brain. Decision surfaces can be used to understand mental diseases. However, improving the performance of prediction is so hard because task-based fMRI datasets can be considered as the imbalanced classification problems. For instance, consider collected data with ten same size categories. Since this dataset is imbalance for (one-versus-all) binary classification, most of the classical algorithms cannot provide acceptable performance. This section proposes IABC framework for decoding different stimuli in the human brain. This framework uses a new binary imbalance AdaBoost algorithm as binary classification approach. It can increase the performance of prediction by exploiting a supervised random sampling and the correlation between classes. In addition, this algorithm is utilized in an Error-Correcting Output Codes (ECOC) method for multi-class prediction of the brain responses. Empirical studies show the superiority of our proposed method in comparison with state-of-the-art approaches.

Chapter 6. Conclusion

One of the greatest challenges in our century is understanding how the human brain works. As an interdisciplinary field of study, computational neuroscience can break neural codes by employing different concepts from the mathematics, physics, psychology, psychiatry, and machine learning. In this thesis, we focus on developing modern machine learning approaches for the different levels of analyzing the neural activities. We first discuss the current challenges for decoding the human brain and then introduce novel techniques for improving this procedure. These techniques have a wide range of real-world applications from exploring novel treatments for mental diseases to creating a new generation of the user interface.

In Chapter 2, we proposed two techniques for functional alignment. Firstly, we develop Deep Hyperalignment (DHA) as a novel unsupervised approach for functional alignment. DHA is not limited by a restricted fixed mapping function because the kernel in DHA is a multi-layer neural network that can separately implement *any nonlinear function*. As a supervised alternative, Local Discriminant Hyperalignment (LDHA) method is introduced, where it incorporates the idea of Local Discriminate Analysis (LDA) into CCA for improving the performance of the hyperalignment solution. In this chapter, we illustrate that while DHA can improve the performance of binary analysis, LDHA depicts better performance for multi-class datasets.

In Chapter 3, we propose novel feature analysis techniques. In fact, the proposed approach estimates a snapshot of brain image for each stimulus rather than analyzing whole of the time series. While the classical methods just can extract features from voxel space, the proposed method selects a subset of time-points for analyzing the neural activities. In practice, these snapshots are selected by finding local maximums in the smoothed version of the design matrix. Finally, we propose two learning approaches. Indeed, extracted features can be analyzed by using both unsupervised learning and supervised learning. This thesis proposed a cluster ensemble approach in order to apply unsupervised learning, where similarities or distances between neural activities can be compared across subjects. As the supervised alternative, we develop a bagging technique by using binary $\ell1$-regularized SVM classifiers, where they are generated by utilizing each of neural activities in the level of anatomical regions. The main contribution of this method is that it can decrease the sparsity of fMRI datasets.

In Chapter 4, we introduce Deep Representational Similarity Analysis (DRSA) as a novel deep extension of RSA method to analyze similarity for both within a dataset, and across different datasets. Like DHA, DRSA also employs a deep network as the kernel function that maps nonlinear neural activities to a linear information-rich embedded space and then evaluates the similarities (or distances) between the mapped features. In addition, DRSA uses a new regularization term for making a trade-off between correlation and covariance of distinctive cognitive tasks. Since DRSA uses gradient-based optimization approaches, it is time efficient for evaluating high-dimensional fMRI images, such as whole-brain datasets. In the end, we evaluate the similarity of different datasets in this chapter, where the datasets related to each type of cognitive tasks (visual stimuli, decision making, and flavor) have more within-category similarity in comparison with distinctive cognitive categories

In Chapter 5, we develop a modified version of imbalance Adapting Boosting (AdaBoost) algorithm for binary classification. This algorithm uses a supervised random sampling and penalty values, which are calculated by the correlation between different classes, for improving the performance of prediction. Indeed, this method is well-suited for one-vs-all classification analysis. Then, we focus on multi-class learning approach. Here, we utilize Error-Correcting Output Codes (ECOC) as an indirect multi-class approach in order to extend the proposed binary classifiers for the multi-class prediction. Empirical studies show the superiority of our proposed methods in comparison with state-of-the-art learning approaches.

Besides the theories and the empirical studies in the thesis, we also make our research easily reproducible and open to the public. We have created a GUI-based toolbox for running the standard pipeline of analyzing task-based fMRI images, including the proposed methods in this thesis, that is available at `https://easyfmri.github.io`. Moreover, we have also prepared a data repository for sharing task-based fMRI datasets. This repository is available at `https://easyfmridata.github.io`.

References

[1] James V Haxby, Andrew C Connolly, and J Swaroop Guntupalli. “Decoding neural representational spaces using multivariate pattern analysis”. In: *Annual Review of Neuroscience* 37 (2014), pp. 435–456.

[2] Po-Hsuan Chen. “Multi-view Representation Learning with Applications to Functional Neuroimaging Data”. Ph.D. Thesis. Princeton University, 2017.

[3] Ming Bo Cai, Nicolas W Schuck, Jonathan W Pillow, and Yael Niv. “A Bayesian method for reducing bias in neural representational similarity analysis”. In: *Advances in Neural Information Processing Systems (NIPS’16)*. 2016, pp. 4951–4959.

[4] Alexander Lorbert and Peter J Ramadge. “Kernel hyperalignment”. In: *Advances in Neural Information Processing Systems*. 2012, pp. 1790–1798.

[5] Alexander Lorbert. “Alignment and Supervised Learning with Functional Neuroimaging Data”. Ph.D. Thesis. Princeton University, 2012.

[6] James V Haxby, J Swaroop Guntupalli, Andrew C Connolly, Yaroslav O Halchenko, Bryan R Conroy, M Ida Gobbini, Michael Hanke, and Peter J Ramadge. “A common, high-dimensional model of the representational space in human ventral temporal cortex”. In: *Neuron* 72.2 (2011), pp. 404–416.

[7] Nikolaus Kriegeskorte, Rainer Goebel, and Peter Bandettini. “Information-based functional brain mapping”. In: *Proceedings of the National academy of Sciences of the United States of America* 103.10 (2006), pp. 3863–3868.

[8] Muhammad Yousefnezhad and Daoqiang Zhang. “Multi-Region Neural Representation: A novel model for decoding visual stimuli in human brains”. In: *Proceedings of the 2017 SIAM International Conference on Data Mining (SDM)*. SIAM. 2017, pp. 54–62.

[9] Po Hsuan Cameron Chen, Janice Chen, Yaara Yeshurun, Uri Hasson, James Haxby, and Peter J Ramadge. “A reduced-dimension fMRI shared response model”. In: *Advances in Neural Information Processing Systems (NIPS’15)*. 2015, pp. 460–468.

[10] Bryan Conroy, Ben Singer, James Haxby, and Peter J Ramadge. “fMRI-based inter-subject cortical alignment using functional connectivity”. In: *Advances in Neural Information Processing Systems*. 2009, pp. 378–386.

[11] Hao Xu, Alexander Lorbert, Peter J Ramadge, J Swaroop Guntupalli, and James V Haxby. “Regularized hyperalignment of multi-set fMRI data”. In: *IEEE Statistical Signal Processing Workshop (SSP)*. IEEE. 2012, pp. 229–232.

[12] Po Hsuan Chen, J Swaroop Guntupalli, James V Haxby, and Peter J Ramadge. “Joint SVD-Hyperalignment for multi-subject FMRI data alignment”. In: *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE. 2014, pp. 1–6.

[13] Muhammad Yousefnezhad and Daoqiang Zhang. “Deep Hyperalignment”. In: *Advances in Neural Information Processing Systems (NIPS)*. 2017, pp. 1603–1611.

[14] Muhammad Yousefnezhad and Daoqiang Zhang. “Local Discriminant Hyperalignment for Multi-Subject fMRI Data Alignment.” In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*. 2017, pp. 59–65.

[15] J Swaroop Guntupalli, Michael Hanke, Yaroslav O Halchenko, Andrew C Connolly, Peter J Ramadge, and James V Haxby. “A model of representational spaces in human cortex”. In: *Cerebral Cortex* (2016), bhw068.

[16] Po-Hsuan Chen, Xia Zhu, Hejia Zhang, Javier S Turek, Janice Chen, Theodore L Willke, Uri Hasson, and Peter J Ramadge. “A convolutional autoencoder for multi-subject fmri data aggregation”. In: *29th Workshop of Representation Learning in Artificial and Biological Neural Networks* (2016).

[17] Michael J Anderson, Mihai Capota, Javier S Turek, Xia Zhu, Theodore L Willke, Yida Wang, Po-Hsuan Chen, Jeremy R Manning, Peter J Ramadge, and Kenneth A Norman. “Enabling factor analysis on thousand-subject neuroimaging datasets”. In: *IEEE International Conference on Big Data*. IEEE. 2016, pp. 1151–1160.

[18] Javier S Turek, Theodore L Willke, Po-Hsuan Chen, and Peter J Ramadge. “A semi-supervised method for multi-subject fMRI functional alignment”. In: *The 41st International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. March/20–25, Shanghai, China: IEEE, 2016.

[19] Muhammad Yousefnezhad and Daoqiang Zhang. “Anatomical Pattern Analysis for decoding visual stimuli in human brains”. In: *Cognitive Computation* (2017), pp. 1–12.

[20] Muhammad Yousefnezhad and Daoqiang Zhang. “Decoding visual stimuli in human brain by using Anatomical Pattern Analysis on fMRI images”. In: *International Conference on Brain Inspired Cognitive Systems*. Springer. 2016, pp. 47–57.

[21] Alice J O’toole, Fang Jiang, Hervé Abdi, and James V Haxby. “Partially distributed representations of objects and faces in ventral temporal cortex”. In: *Journal of Cognitive Neuroscience* 17.4 (2005), pp. 580–590.

[22] Jiansong Xu, Marc N Potenza, and Vince D Calhoun. “Spatial ICA reveals functional activity hidden from traditional fMRI GLM-based analyses”. In: *Frontiers in Neuroscience* 7 (2013).

[23] Michael Anderson and Tim Oates. “A critique of multi-voxel pattern analysis”. In: *Proceedings of the Cognitive Science Society*. Vol. 32. 2010.

[24] Holger Mohr, Uta Wolfensteller, Steffi Frimmel, and Hannes Ruge. “Sparse regularization techniques provide novel insights into outcome integration processes”. In: *NeuroImage* 104 (2015), pp. 163–176.

[25] Brenton W McMenamin, Rebecca G Deason, Vaughn R Steele, Wilma Koutstaal, and Chad J Marsolek. “Separability of abstract-category and specific-exemplar visual object subsystems: Evidence from fMRI pattern analysis”. In: *Brain and Cognition* 93 (2015), pp. 54–63.

[26] Stephen José Hanson, Toshihiko Matsuka, and James V Haxby. “Combinatorial codes in ventral temporal lobe for object recognition: Haxby (2001) revisited: is there a “face” area?” In: *NeuroImage* 23.1 (2004), pp. 156–166.

[27] Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. “Representational similarity analysis–connecting the branches of systems neuroscience”. In: *Frontiers in Systems Neuroscience* 2 (2008).

[28] Andrew C Connolly, J Swaroop Guntupalli, Jason Gors, Michael Hanke, Yaroslav O Halchenko, Yu-Chien Wu, Herve Abdi, and James V Haxby. “The representation of biological classes in the human brain”. In: *Journal of Neuroscience* 32.8 (2012), pp. 2608–2618.

[29] Chris Hans. “Bayesian lasso regression”. In: *Biometrika* 96.4 (2009), pp. 835–845.

[30] Arthur E Hoerl and Robert W Kennard. “Ridge regression: Biased estimation for nonorthogonal problems”. In: *Technometrics* 12.1 (1970), pp. 55–67.

[31] Ming Yuan and Yi Lin. “Model selection and estimation in regression with grouped variables”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.1 (2006), pp. 49–67.

[32] Hui Zou and Trevor Hastie. “Regularization and variable selection via the elastic net”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2 (2005), pp. 301–320.

[33] Howard D Bondell and Brian J Reich. “Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR”. In: *Biometrics* 64.1 (2008), pp. 115–123.

[34] Urvashi Oswal, Christopher Cox, Matthew Lambon-Ralph, Timothy Rogers, and Robert Nowak. “Representational similarity learning with application to brain networks”. In: *International Conference on Machine Learning (ICML)*. 2016, pp. 1041–1049.

[35] Mario Figueiredo and Robert Nowak. “Ordered weighted l1 regularized regression with strongly correlated covariates: Theoretical aspects”. In: *Artificial Intelligence and Statistics (AAAI)*. 2016, pp. 930–938.

[36] Muhammad Yousefnezhad and Daoqiang Zhang. “Weighted spectral cluster ensemble”. In: *Data Mining (ICDM), 2015 IEEE International Conference on*. IEEE. 2015, pp. 549–558.

[37] Muhammad Yousefnezhad, Ali Reihanian, Daoqiang Zhang, and Behrouz Minaei-Bidgoli. “A new selection strategy for selective cluster ensemble based on Diversity and Independency”. In: *Engineering Applications of Artificial Intelligence* 56 (2016), pp. 260–272.

[38] Muhammad Yousefnezhad, Sheng-Jun Huang, and Daoqiang Zhang. “WoCE: a framework for clustering ensemble by exploiting the wisdom of Crowds theory”. In: *IEEE Transactions on Cybernetics* 48.2 (2017), pp. 486–499.

[39] AM Clare Kelly, Lucina Q Uddin, Bharat B Biswal, F Xavier Castellanos, and Michael P Milham. “Competition between functional brain networks mediates behavioral variability”. In: *NeuroImage* 39.1 (2008), pp. 527–537.

[40] Yan Peng, Daoqiang Zhang, and Jianchun Zhang. “A new canonical correlation analysis algorithm with local discrimination”. In: *Neural Processing Letters* 31.1 (2010), pp. 1–15.

[41] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. “Deep Canonical Correlation Analysis”. In: *30th International Conference on Machine Learning (ICML’13)*. Vol. 28. June/16–21, Atlanta, USA, 2013, pp. 1247–1255.

[42] Adrian Benton, Huda Khayrallah, Biman Gujral, Drew Reisinger, Sheng Zhang, and Raman Arora. “Deep Generalized Canonical Correlation Analysis”. In: *5th International Conference on Learning Representations (ICLR)*. April/24–26, Toulon, France, 2017.

[43] Jean Talairach and Pierre Tournoux. *Co-planar stereotaxic atlas of the human brain. 3-Dimensional proportional system: an approach to cerebral imaging*. Thieme, 1988.

[44] Alan C Evans, D Louis Collins, SR Mills, ED Brown, RL Kelly, and Terry M Peters. “3D statistical neuroanatomical models from 305 MRI volumes”. In: *Nuclear Science Symposium and Medical Imaging Conference, 1993., 1993 IEEE Conference Record.* IEEE. 1993, pp. 1813–1817.

[45] Wilkin Chau and Anthony R McIntosh. “The Talairach coordinate of a point in the MNI space: how to interpret it”. In: *NeuroImage* 25.2 (2005), pp. 408–416.

[46] John DG Watson, Ralph Myers, Richard S J Frackowiak, Joseph V Hajnal, Roger P Woods, John C Mazziotta, Stewart Shipp, and Semir Zeki. “Area V5 of the human brain: evidence from a combined study using positron emission tomography and magnetic resonance imaging”. In: *Cerebral Cortex* 3.2 (1993), pp. 79–94.

[47] J Rademacher, V S Caviness, H Steinmetz, and AM Galaburda. “Topographical variation of the human primary cortices: implications for neuroimaging, brain mapping, and neurobiology”. In: *Cerebral Cortex* 3.4 (1993), pp. 313–329.

[48] Mert R Sabuncu, Benjamin D Singer, Bryan Conroy, Ronald E Bryan, Peter J Ramadge, and James V Haxby. “Function-based intersubject alignment of human cortical anatomy”. In: *Cerebral Cortex* 20.1 (2010), pp. 130–140.

[49] Jacek P Dmochowski, Paul Sajda, Joao Dias, and Lucas C Parra. “Correlated components of ongoing EEG point to emotionally laden attention–a possible marker of engagement?” In: *Frontiers in Human Neuroscience* 6 (2012), p. 112.

[50] Andrew M Michael, Mathew Anderson, Robyn L Miller, Tülay Adalı, and Vince D Calhoun. "Preserving subject variability in group fMRI analysis: performance evaluation of GICA vs. IVA". In: *Distributed Networks-New Outlooks on Cerebellar Function* 106 (2015), pp. 1–18.

[51] John C Gower and Garmt B Dijksterhuis. *Procrustes problems*. Vol. 30. Oxford University Press on Demand, 2004.

[52] Jing Sui, Godfrey Pearlson, Arvind Caprihan, Tülay Adali, Kent A Kiehl, Jingyu Liu, Jeremy Yamamoto, and Vince D Calhoun. "Discriminating schizophrenia and bipolar disorder by fusing fMRI and DTI in a multimodal CCA+ joint ICA model". In: *NeuroImage* 57.3 (2011), pp. 839–855.

[53] Jing Sui, Hao He, Godfrey D Pearlson, Tülay Adali, Kent A Kiehl, Qingbao Yu, Vince P Clark, Eduardo Castro, Tonya White, Bryon A Mueller, et al. "Three-way (N-way) fusion of brain imaging data based on mCCA+ jICA and its application to discriminating schizophrenia". In: *NeuroImage* 66 (2013), pp. 119–132.

[54] Pushpendre Rastogi, Benjamin Van Durme, and Raman Arora. "Multiview LSA: Representation learning via generalized CCA". In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2015, pp. 556–566.

[55] Matthew Brand. "Incremental singular value decomposition of uncertain data with missing values". In: *European Conference on Computer Vision*. Springer. 2002, pp. 707–720.

[56] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". In: *Nature* 323.6088 (1986), p. 533.

[57] Alex J Smola and Bernhard Schölkopf. "A tutorial on support vector regression". In: *Statistics and Computing* 14.3 (2004), pp. 199–222.

[58] Xianchao Zhang, Long Zhao, Linlin Zong, and Xinyue Liu. "Multi-View Clustering via Multi-Manifold Regularized Nonnegative Matrix Factorization". In: *IEEE International Conference on Data Mining series (ICDM'14)*. Shenzhen, China, 2014.

[59] A. Strehl and J. Ghosh. "Cluster ensembles - a knowledge reuse framework for combining multiple partitions". In: *Journal of Machine Learning Research* 3 (2002), pp. 583–617.

[60] A. Fred and A. Lourenco. "Cluster Ensemble Methods: from Single Clusterings to Combined Solutions". In: *Computer Intelligence* 126 (2008), pp. 3–30.

[61] A. Fred and A. K. Jain. "Combining Multiple Clusterings Using Evidence Accumulation". In: *IEEE Transaction on Pattern Analysis and Machine Intelligence* 27 (2005), pp. 835–850.

[62] X. Fern and W. Lin. "Cluster Ensemble Selection". In: *SIAM International Conference on Data Mining (SDM'08)*. Atlanta, Georgia, USA: SIAM, 2008, pp. 128–141.

[63] H. Alizadeh, B. Minaei-Bidgoli, and H. Parvin. "Cluster Ensemble Selection Based on a New Cluster Stability Measure". In: *Intelligence Data Analysis (IDA)* 18.3 (2014), pp. 389–40.

[64] H. Alizadeh, M. Yousefnezhad, and B. Minaei-Bidgoli. "Wisdom of Crowds Cluster Ensemble". In: *Intelligent Data Analysis (IDA)* 19.3 (2015).

[65] J. Azimi and X. Fern. "Adaptive cluster ensemble selection". In: *21th International joint conference on artificial intelligence (IJCAI-09)*. Pasadena, CA, USA, 2009, pp. 992–997.

[66] J. Jia, X. Xiao, and B. Liu. "Similarity-based Spectral Clustering Ensemble Selection". In: *9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*. Jingdezhen Ceramic Institute, Jingdezhen, China, 2012, pp. 1071–1074.

[67] S. Romano, J. Bailey, N. X. Vinh, and K. Verspoor. "Standardized Mutual Information for Clustering Comparisons: One Step Further in Adjustment for Chance". In: *31st International Conference on Machine Learning (ICML'14)*. Beijing, China, 2014, pp. 1143–1151.

[68] R Baumgartner, R Somorjai, R Summers, W Richter, L Ryner, and M Jarmasz. “Resampling as a cluster validation technique in fMRI”. In: *Journal of Magnetic Resonance Imaging* 11.2 (2000), pp. 228–231.

[69] João R Sato, André Fujita, Elisson F Cardoso, Carlos E Thomaz, Michael J Brammer, and Edson Amaro Jr. “Analyzing the connectivity between regions of interest: an approach based on cluster Granger causality for fMRI data analysis”. In: *NeuroImage* 52.4 (2010), pp. 1444–1455.

[70] Pierre Bellec, Pedro Rosa-Neto, Oliver C Lyttelton, Habib Benali, and Alan C Evans. “Multi-level bootstrap analysis of stable clusters in resting-state fMRI”. In: *NeuroImage* 51.3 (2010), pp. 1126–1139.

[71] Paola Galdi, Michele Fratello, Francesca Trojsi, Antonio Russo, Gioacchino Tedeschi, Roberto Tagliaferri, and Fabrizio Esposito. “Consensus-based feature extraction in rs-fMRI data analysis”. In: *Soft Computing* 22.11 (2018), pp. 3785–3795.

[72] Chia-Tung Kuo, Peter Walker, Owen Carmichael, and Ian Davidson. “Spectral Clustering for Medical Imaging”. In: *IEEE International Conference on Data Mining series (ICDM’14)*. Shenzhen, China, 2014.

[73] J. Gao, F. Liang, W. Fan, Y. Sun, and J. Han. “A Graph-based Consensus Maximization Approach for Combining Multiple Supervised and Unsupervised Models”. In: *IEEE Transactions on Knowledge and Data Engineering* 25.1 (2013), pp. 15–2.

[74] Rafael Malach, JB Reppas, RR Benson, KK Kwong, H Jiang, WA Kennedy, PJ Ledden, TJ Brady, BR Rosen, and RB Tootell. “Object-related activity revealed by functional magnetic resonance imaging in human occipital cortex”. In: *Proceedings of the National Academy of Sciences (PNAS)* 92.18 (1995), pp. 8135–8139.

[75] Nancy Kanwisher, Josh McDermott, and Marvin M Chun. “The fusiform face area: a module in human extrastriate cortex specialized for face perception”. In: *Journal of Neuroscience* 17.11 (1997), pp. 4302–4311.

[76] Laurent Cohen, Stanislas Dehaene, Lionel Naccache, Stéphane Lehéricy, Ghislaine Dehaene-Lambertz, Marie-Anne Hénaff, and François Michel. “The visual word form area: spatial and temporal characterization of an initial stage of reading in normal subjects and posterior split-brain patients”. In: *Brain* 123.2 (2000), pp. 291–307.

[77] Paul E Downing, Yuhong Jiang, Miles Shuman, and Nancy Kanwisher. “A cortical area selective for visual processing of the human body”. In: *Science* 293.5539 (2001), pp. 2470–2473.

[78] James V Haxby, M Ida Gobbini, Maura L Furey, Alumit Ishai, Jennifer L Schouten, and Pietro Pietrini. “Distributed and overlapping representations of faces and objects in ventral temporal cortex”. In: *Science* 293.5539 (2001), pp. 2425–2430.

[79] Yukiyasu Kamitani and Frank Tong. “Decoding the visual and subjective contents of the human brain”. In: *Nature Neuroscience* 8.5 (2005), pp. 679–685.

[80] John-Dylan Haynes and Geraint Rees. “Decoding mental states from brain activity in humans”. In: *Nature Reviews Neuroscience* 7.7 (2006), p. 523.

[81] John-Dylan Haynes, Katsuyuki Sakai, Geraint Rees, Sam Gilbert, Chris Frith, and Richard E Passingham. “Reading hidden intentions in the human brain”. In: *Current Biology* 17.4 (2007), pp. 323–328.

[82] Grace E Rice, David M Watson, Tom Hartley, and Timothy J Andrews. “Low-level image properties of visual objects predict patterns of neural response across category-selective regions of the ventral visual pathway”. In: *Journal of Neuroscience* 34.26 (2014), pp. 8837–8844.

[83] David D Cox and Robert L Savoy. “Functional magnetic resonance imaging (fMRI) ‘brain reading’: detecting and classifying distributed patterns of fMRI activity in human visual cortex”. In: *NeuroImage* 19.2 (2003), pp. 261–270.

[84] Thomas A Carlson, Paul Schrater, and Sheng He. “Patterns of activity in the categorical representations of objects”. In: *Journal of Cognitive Neuroscience* 15.5 (2003), pp. 704–717.

[85] Kenneth A Norman, Sean M Polyn, Greg J Detre, and James V Haxby. “Beyond mind-reading: multi-voxel pattern analysis of fMRI data”. In: *Trends in Cognitive Sciences* 10.9 (2006), pp. 424–430.

[86] Kendrick N Kay, Thomas Naselaris, Ryan J Prenger, and Jack L Gallant. “Identifying natural images from human brain activity”. In: *Nature* 452.7185 (2008), p. 352.

[87] Tom M Mitchell, Svetlana V Shinkareva, Andrew Carlson, Kai-Min Chang, Vicente L Malave, Robert A Mason, and Marcel Adam Just. “Predicting human brain activity associated with the meanings of nouns”. In: *Science* 320.5880 (2008), pp. 1191–1195.

[88] Yoichi Miyawaki, Hajime Uchida, Okito Yamashita, Masa-aki Sato, Yusuke Morito, Hiroki C Tanabe, Norihiro Sadato, and Yukiyasu Kamitani. “Visual image reconstruction from human brain activity using a combination of multiscale local image decoders”. In: *Neuron* 60.5 (2008), pp. 915–929.

[89] Nikolaus Kriegeskorte, W Kyle Simmons, Patrick SF Bellgowan, and Chris I Baker. “Circular analysis in systems neuroscience: the dangers of double dipping”. In: *Nature Neuroscience* 12.5 (2009), pp. 535–540.

[90] Okito Yamashita, Masa-aki Sato, Taku Yoshioka, Frank Tong, and Yukiyasu Kamitani. “Sparse estimation automatically selects voxels relevant for the decoding of fMRI activity patterns”. In: *NeuroImage* 42.4 (2008), pp. 1414–1429.

[91] Melissa K Carroll, Guillermo A Cecchi, Irina Rish, Rahul Garg, and A Ravishankar Rao. “Prediction and interpretation of distributed neural activity with sparse models”. In: *NeuroImage* 44.1 (2009), pp. 112–122.

[92] Gael Varoquaux, Alexandre Gramfort, and Bertrand Thirion. “Small-sample brain mapping: sparse recovery on spatially correlated designs with randomization and clustering”. In: *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*. 2012, pp. 1375–1382.

[93] David E Osher, Rebecca R Saxe, Kami Koldewyn, John DE Gabrieli, Nancy Kanwisher, and Zeynep M Saygin. “Structural connectivity fingerprints predict cortical selectivity for multiple visual categories across cortex”. In: *Cerebral Cortex* 26.4 (2015), pp. 1668–1683.

[94] Karl J Friston. “Statistical parametric mapping”. In: *Neuroscience Databases*. Springer, 2003, pp. 237–250.

[95] Mark Jenkinson, Peter Bannister, Michael Brady, and Stephen Smith. “Improved optimization for the robust and accurate linear registration and motion correction of brain images”. In: *NeuroImage* 17.2 (2002), pp. 825–841.

[96] A. Ng, M. Jordan, and Y. Weiss. “On Spectral Clustering: Analysis and an algorithm”. In: *Advances in Neural Information Processing Systems 14 (NIPS’01)*. MIT Press, 2001, pp. 849–856.

[97] D. Yan, L. Huang, and M. I. Jordan. “Fast approximate spectral clustering”. In: *15th ACM Conference on Knowledge Discovery and Data Mining (SIGKDD)*. Paris, France, 2009.

[98] A. Clauset, M. Newman, and C. Moore. “Finding community structure in very large networks”. In: *Physical Review E* 70.066111 (2004).

[99] M. E. J. Newman. “Modularity and community structure in networks”. In: *Proceedings of the National Academy of Sciences of the United States of America* 103.23 (2006), pp. 8577–8696.

[100] Paul S Bradley and Olvi L Mangasarian. “Feature selection via concave minimization and support vector machines.” In: *ICML*. Vol. 98. 1998, pp. 82–90.

[101] Corinna Cortes and Vladimir Vapnik. “Support-vector networks”. In: *Machine learning* 20.3 (1995), pp. 273–297.

[102] Leo Breiman. “Bagging predictors”. In: *Machine learning* 24.2 (1996), pp. 123–140.

[103] Kevin P Murphy. *Machine Learning: A Probabilistic Perspective. Adaptive Computation and Machine Learning*. 2012.

[104] Seyed-Mahdi Khaligh-Razavi and Nikolaus Kriegeskorte. “Deep supervised, but not unsupervised, models may explain IT cortical representation”. In: *PLoS Computational Biology* 10.11 (2014), e1003915.

[105] Seyed-Mahdi Khaligh-Razavi, Linda Henriksson, Kendrick Kay, and Nikolaus Kriegeskorte. “Fixed versus mixed RSA: Explaining visual representations by fixed and mixed feature sets from shallow and deep computational models”. In: *Journal of Mathematical Psychology* 76 (2017), pp. 184–197.

[106] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *International Conference on Learning Representations (ICLR)*. 2015.

[107] Krzysztof C Kiwiel. “Convergence and efficiency of subgradient methods for quasiconvex minimization”. In: *Mathematical Programming* 90.1 (2001), pp. 1–25.

[108] Keith J Duncan, Chotiga Pattamadilok, Iris Knierim, and Joseph T Devlin. “Consistency and variability in functional localisers”. In: *NeuroImage* 46.4 (2009), pp. 1018–1026.

[109] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. “Exploratory undersampling for class-imbalance learning”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39.2 (2009), pp. 539–550.

[110] Sergio Escalera, Oriol Pujol, and Petia Radeva. “Error-correcting output codes library”. In: *Journal of Machine Learning Research* 11.Feb (2010), pp. 661–664.

[111] Tom Schonberg, Craig R Fox, Jeanette A Mumford, Eliza Congdon, Christopher Trepel, and Russell A Poldrack. “Decreasing ventromedial prefrontal cortex activity during sequential risk-taking: an fMRI investigation of the balloon analog risk task”. In: *Frontiers in Neuroscience* 6 (2012), p. 80.

[112] Adam R Aron, Mark A Gluck, and Russell A Poldrack. “Long-term test–retest reliability of functional MRI in a classification learning task”. In: *NeuroImage* 29.3 (2006), pp. 1000–1006.

[113] Sabrina M Tom, Craig R Fox, Christopher Trepel, and Russell A Poldrack. “The neural basis of loss aversion in decision-making under risk”. In: *Science* 315.5811 (2007), pp. 515–518.

[114] Karin Foerde, Barbara J Knowlton, and Russell A Poldrack. “Modulation of competing memory systems by distraction”. In: *Proceedings of the National Academy of Sciences* 103.31 (2006), pp. 11778–11783.

[115] Russell A Poldrack, Jill Clark, EJ Pare-Blagoev, Daphna Shohamy, J Creso Moyano, Catherine Myers, and Mark A Gluck. “Interactive memory systems in the human brain”. In: *Nature* 414.6863 (2001), p. 546.

[116] Jennifer M Walz, Robin I Goldman, Michael Carapezza, Jordan Muraskin, Truman R Brown, and Paul Sajda. “Simultaneous EEG-fMRI reveals temporal evolution of coupling between supramodal cortical attention networks and the brainstem”. In: *Journal of Neuroscience* 33.49 (2013), pp. 19212–19222.

[117] Timothy D Verstynen. “The organization and dynamics of corticostriatal pathways link the medial orbitofrontal cortex to future behavioral responses”. In: *Journal of Neurophysiology* 112.10 (2014), pp. 2457–2469.

[118] Maria Geraldine Veldhuizen, Richard Keith Babbs, Barkha Patel, Wambura Fobbs, Nils B Kroemer, Elizabeth Garcia, Martin R Yeomans, and Dana M Small. “Integration of Sweet Taste and Metabolism Determines Carbohydrate Reward”. In: *Current Biology* 27.16 (2017), pp. 2476–2485.

[119] Krzysztof J Gorgolewski, Tibor Auer, Vince D Calhoun, R Cameron Craddock, Samir Das, Eugene P Duff, Guillaume Flandin, Satrajit S Ghosh, Tristan Glatard, Yaroslav O Halchenko, et al. “The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments”. In: *Scientific Data* 3 (2016), p. 160044.

[120] Mark Jenkinson and Stephen Smith. “A global optimisation method for robust affine registration of brain images”. In: *Medical Image Analysis* 5.2 (2001), pp. 143–156.

[121] Stephen M Smith. “Fast robust automated brain extraction”. In: *Human Brain Mapping* 17.3 (2002), pp. 143–155.

Acknowledgments

Ph.D. studies is a long but gratifying experience for me. During the past four years, I have learned and grown so much in both aspects, i.e., academic careers and personal life. Indeed, it made me a much more mature person as well as a much more independent researcher. This thesis is one of the best examples to shed light my career developments.

I would like to express my highest appreciation to the best Ph.D. supervisor in the world in point of my view, Prof. Daoqiang Zhang. I owe a great debt of gratitude to him for his generous support and guidance during my Ph.D. studies. From the beginning, he trusted me and provided much freedom to study different issues and discover my interested directions. He always suggests me effective feedback all along the exploration. Through the past four years, I have learned from him how to be honest when facing a problem and how to deal with an issue by using a down-to-earth approach. I have always enjoyed our weekly meetings.

I could not be able to reach the current stage without the support from several committees that I have worked with along the way. I would like to thank Prof. Songcan Chen, Dr. Sheng-Jun Huang, and also the ‘easy fMRI:data’ team members, including Tonglin Xu, Xiaoliang Sheng, Shuo Huang, Weida Li, and Ali Rawashdeh. Further, I would like to thank Shao Wei, my closest friend in China, for his kind support.

In the end, I want to thank the staff in the Nanjing University of Aeronautics and Astronautics for their enormous help.

Thanks everybody :)

Publications

We have published following papers during this study:

Conference Papers

1. **Muhammad Yousefnezhad** and Daoqiang Zhang. 'Deep Hyperalignment'. In: *Advances in Neural Information Processing Systems (NIPS)*. 2017, pp. 1603–1611.

2. **Muhammad Yousefnezhad** and Daoqiang Zhang. 'Multi-Region Neural Representation: A novel model for decoding visual stimuli in human brains'. In: *Proceedings of the 2017 SIAM International Conference on Data Mining (SDM)* SIAM. 2017, pp. 54–62.

3. **Muhammad Yousefnezhad** and Daoqiang Zhang. 'Local Discriminant Hyperalignment for Multi-Subject fMRI Data Alignment'. In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*. 2017, pp. 59–65.

4. **Muhammad Yousefnezhad** and Daoqiang Zhang. 'Decoding visual stimuli in human brain by using Anatomical Pattern Analysis on fMRI images'. In: *International Conference on Brain Inspired Cognitive Systems*. Springer. 2016, pp. 47–57.

5. **Muhammad Yousefnezhad** and Daoqiang Zhang. 'Weighted spectral cluster ensemble'. In: *International Conference on Data Mining (ICDM)*, IEEE. 2015, pp. 549–558.

Journal Papers

1. **Muhammad Yousefnezhad** and Daoqiang Zhang. 'Anatomical Pattern Analysis for decoding visual stimuli in human brains'. In: *Cognitive Computation*. (2017), pp. 1–12.

2. **Muhammad Yousefnezhad**, Sheng-Jun Huang, and Daoqiang Zhang. 'WoCE: a framework for clustering ensemble by exploiting the wisdom of Crowds theory'. In: *IEEE Transactions on Cybernetics* 48.2 (2017), pp. 486–499.

3. **Muhammad Yousefnezhad**, Ali Reihanian, Daoqiang Zhang, Behrouz Minaei-Bidgoli 'A new selection strategy for selective cluster ensemble based on Diversity and Independency'. In: *Engineering Applications of Artificial Intelligence* 56 (2016), pp. 260–272.

攻读博士学位期间参加科研项目情况

1. 国家优秀青年科学基金“脑影像智能计算”(61422204)，参与

2. 国家自然科学基金“属性学习及其应用研究”(61473149)，参与

3. 江苏省自然科学基金杰出青年基金“基于机器学习的多模态神经影像分析及应用”(BK20130034)，参与

4. 南京航空航天大学杰出人才培育基金“机器学习与智能影像分析”(NE201305)，参与

Appendix A. Datasets

In this section, we introduce datasets that are employed in this thesis. Here, we use two group of datasets, i.e., Region of Interests (ROI) based data, and whole-based datasets. Indeed, we analyze some parts of brain images in ROI-based data, where these parts are manually selected based on the original papers of each data. In this thesis, we use 'R' prefix for the ROI-based dataset. By contrast, whole-brain datasets include all of the neural activities in the brains, where the images are registered to a standard space, i.e., Montreal Neurological Institute (MNI) 152 space $T1$ with voxel size $4mm$ [19]. Further, a 'W' prefix is used for denoting whole-brain data. While the ROI-images enable us to analyze the neural activities in the specific loci, whole-brain data can be used to figure out what information is represented in a region of the human brain and how that information is encoded. Since the number of selected features (voxels) in the ROI is significantly smaller than the whole of voxels in a fMRI image, the runtime of algorithms by utilizing ROI-based datasets are rapidly faster than whole brain datasets.

Table 1 illustrates the technical information of the employed datasets. In dataset 001, subjects perform the Balloon Analog Risk-taking Task in an event-related design, where we analyze the whole-brain neural activities. For more information, please see [111]. In dataset 002, subjects performed a classification learning task with two different problems, using a 'weather prediction' task. In the probabilistic problem (002P), the labels were probabilistically related to each set of cards. In the deterministic problem (002D), the labels were deterministically related to each set of cards. More technical information can be found in [112]. In data 005, subjects were presented with mixed (gain/loss) gambles and decided whether they would accept each gamble. No outcomes of these gambles were presented during scanning, but after the scan, three gamblers were selected at random and played for real money. Please see [113] for more information. In dataset 011, participants were trained on two different classification problems. To measure how well participants had learned under each condition, without feedback task (denoted by 011W) was presented during the probe block, and all items were presented under single-task conditions. As the next task, subjects learned the categories based on trial-by-trial feedback. After training, subjects received an additional block of probe trials using a

Table 1 The datasets.

Title	ID	Task Type	S	P	T	V_{ROI}	Scan	TR	TE
Balloon Analog Risk [111]	001	decision	16	4	894	–	S	2000	77
Deterministic classification [112]	002D	decision	17	2	356	–	S	2000	20
Probabilistic classification [112]	002P	decision	17	2	356	–	S	2000	20
Mixed-gambles [113]	005	decision	16	4	714	–	S	2000	30
Dual-task weather prediction [114]	011D	decision	14	3	408	–	S	2000	25
Weather prediction without feedback [114]	011W	decision	14	4	236	–	S	2000	25
Selective stop signal task [112]	017	decision	8	6	546	–	S	2000	$\sim$25
Reversal weather prediction [115]	052R	decision	13	2	450	–	S	2000	20
Weather prediction [115]	052W	decision	13	2	450	–	S	2000	20
Flanker task [39]	102	decision	26	2	292	–	S	2000	20
Visual object recognition [78]	105	visual	6	8	1452	2294	G	2500	30
Word and object processing [108]	107	visual	49	4	322	422	S	2000	28
Auditory odd ball [116]	116A	audio	17	2	510	–	P	2000	20
Visual odd ball [116]	116V	visual	17	2	510	–	P	2000	20
Stroop [117]	164	decision	28	2	370	–	S	1500	10
Integration of sweet taste [118]	231	flavor	9	6	1119	–	S	2000	30
Face-coding localizer (objects) task [7]	232	visual	10	4	760	9947	S	1060	16

S is the number of subject; P denotes the number of stimulus categories; T is the number of scans in unites of TRs (Time of Repetition); V_{ROI} denotes the number of voxels in ROI; 19742 voxels are extracted from MNI 152-T1-4mm space [19] for all whole-brain datasets. Scan(ners) include G = General Electric, P=Philips, or S=Siemens in 3 Tesla; TR is Time of Repetition in millisecond; TE denotes Echo Time in millisecond; Please see `https://openfmri.org` for more information.

mixed event-related fMRI paradigm, during which they classified items that had been trained under dual-task conditions (denoted by 011D). More technical information can be explored in [114]. In dataset 017, subjects performed selective stop-signal classification, where the technical information can be found in [112]. In data 052, participants performed two blocks of an event-related probabilistic weather prediction task (denoted by 052W). Next, they performed two more blocks of the same task with the reward contingencies reversed that is defined by 052R. For more information, please see [115]. In dataset 102, subjects performed a slow event-related Eriksen Flanker task. On each trial, participants used one of two buttons on a response pad to indicate the direction of a central arrow in an array of 5 arrows. In congruent trials the flanking arrows pointed in the same direction as the central arrow (e.g., $< < < < <$), while in more demanding incongruent trials the flanking arrows pointed in the opposite direction (e.g., $< < > < <$). Participants performed two 5-minute blocks, each containing 12 congruent and 12 incongruent trials, presented in a pseudorandom order. Technical information can be found in [39]. In data 105, participants watched eight categories of visual stimuli, i.e., faces, houses, cats, bottles, scissors, shoes, chairs, and scramble patterns. The neural activities for this dataset not only is analyzed for whole-brain images but also we collect data from the temporal cortex (VT) as ROI-based data. More information is denoted in [78]. Dataset 107 is also related to the visual cognitive tasks. Here, subjects performed a visual one-back with four categories of items, i.e., written words, objects, scrambled objects and consonant letter strings. Further, ROI for this dataset is selected based on the main reference [108]. In data 116, subjects performed separate but analogous auditory and visual oddball tasks (interleaved), while we recorded simultaneous EEG-fMRI. This thesis has only used fMRI images. For more information, please see [116]. In dataset 164, subjects performed the color-word version of the Stroop task with three conditions, including congruent, incongruent, and neutral. Participants were instructed to ignore the meaning of the printed word and respond to the ink color in which the word was printed. Each condition was meant to elicit a certain level of attentional demand. Participants responded to ink color by pressing a button under the index, middle, and ring fingers on their right hand. One button for each color (red, green, and blue) on an MR-safe response box. More information is denoted in [117]. In dataset 231, non-caloric beverages were mixed with new flavors, citric acid, sucralose and food coloring. Subjects with three similarly liked but differently flavored and colored beverages who were unable to detect maltodextrin participated in six exposure sessions during which each beverage was consumed six times consistently paired with one of three caloric loads (0, 112.5 and 150 kcal). A fMRI session followed in which participants sampled the non-caloric versions of the three exposed beverage (CS-, CS112.5, and CS150), as well as a tasteless and odorless control solution. Technical information can be found at [118]. Finally, data 232 measured human fMRI responses and psychophysical similarity judgments related to four categories, i.e., objects, faces, places, scramble photos. This thesis analyzed the neural activities of dataset 232 in two different forms, i.e., whole-brain data and the fusiform face area (FFA) as ROI region. More information is denoted in [7].

Appendix B. Preprocessing Steps

This section briefly explains how datasets are preprocessed in this thesis. As mentioned before, we have developed a new toolbox called ‘easy fMRI’ for analyzing fMRI datasets (see Figure 1). This toolbox automatically generates preprocessing script for each fMRI session, and then frequently executes these scripts by using ‘FSL 5.0.10’ (available at `https://fsl.fmrib.ox.ac.uk`). As the first step, all files of each dataset are structured in Brain Imaging Data Structure (BIDS) format [119]. Then, we have utilized the technical information (such as TR, FWHM, etc.) extracted from the original paper of each data for generating the preprocessing script in easy fMRI. Finally, our toolbox has applied the preprocessing steps to the raw fMRI images, including anatomical registration, motion correction, non-brain removal, spatial smoothing. Indeed, we have used Linear Image Registration Tool (FLIRT) algorithm [120] to anatomically register fMRI images to the Montreal Neurological Institute (MNI) 152 standard space [44, 45]. After that, MCFLIRT method [95] and BET technique [121] is respectively employed to apply motion correction, and removing non-brain tissue from the raw images. As the final step for preprocessing, fMRI images are smoothed by using a Gaussian kernel of FWHM $5.0mm$; grand-mean intensity normalization. Figure 1 compares raw and preprocessed fMRI images. Further, we have manually applied the temporal alignment, including the same number of time points are selected for all subject in each dataset.

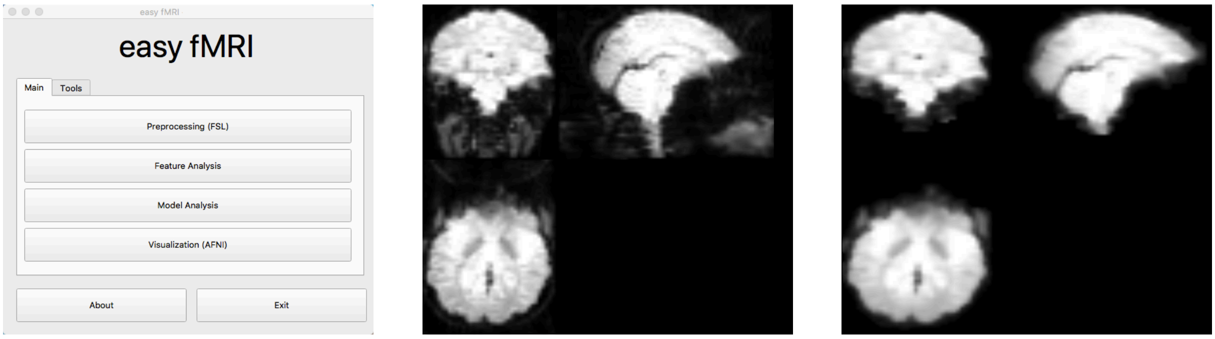


Figure 1 (left) a screenshot of easy fMRI toolbox (middle) A snapshot of raw fMRI image (right) A snapshot of preprocessed fMRI image

Appendix C. 详细中文摘要

目前在神经科学和机器学习领域中的一个重大的挑战是如何正确地理解人类大脑的工作方式，其中对大脑中思想、记忆和情感等功能的研究将促进包括科学、医学、教育等领域的发展。功能性神经影像作为一种成像技术能够用来表示测量到的神经活动，其主要思想是利用这些影像数据来揭示认知过程。事实上，借助这些影像数据，我们不仅能够了解人类大脑区域的功能，而且能够研究这些区域所代表的具体信息以及该信息是如何编码的。神经活动可以从不同层面进行分析，其中最关键的步骤是了解不同的认知任务之间的相似性 (或差异性)，进而建立认知模型来分析神经活动，这能够帮助我们增加对人脑的认知并促进精神疾病治疗的发展。

神经活动可以在不同层面进行分析，但主要步骤是了解分散注意力的认知任务之间的相似性（或差异）。为了测量神经活动，可以使用不同的测量方式，包括事件相关光学信号（Event-Related Optical Signal, EROS），正电子发射断层扫描（Positron Emission Tomography, PET），单光子发射计算机断层扫描（Single-Photon Emission Computed Tomography, SPECT），近红外光谱（Near-Infrared Spectroscopy, NIRS），脑磁图（magnetoencephalography, MEG），脑皮层电图（electrocorticography, ECoG），脑电图（electroencephalography, EEG），功能磁共振成像（functional Magnetic Resonance Imaging, fMRI）。像大多数先前的研究一样，本论文将关注fMRI 图像，该技术通过使用血氧水平依赖（functional Magnetic Resonance Imaging, BOLD）对比作为神经激活的度量来测量神经活动。主要思想是利用神经活动的测量数据来阐明认知过程。实际上，fMRI 使我们能够询问在人类大脑某块区域中表示什么信息以及该区域如何对信息进行编码，而不仅仅是询问区域的功能是什么。使用 fMRI 技术破解神经代码有两个主要原因。首先，它是一种非侵入性成像技术。与其他非侵入性脑成像技术相比，它还具有前所未有的时空分辨率，没有已知的副作用。

神经活动可以以不同的形式进行分析，包括图形结构，连续信号，基于成分的表示等。在fMRI 数据集中，神经活动通常采用体素（脑图像中的体积元素）的形式。人脑解码基础的核心概念是高维和大数据表示空间。例如，被试者的 fMRI 响应（包括感兴趣区域（Region of Interests, ROI）中的六个时间点和 1000 个体素）必须由 6000 维空间中的向量定义。实际上，每个被试者的大脑神经活动被认为是神经代表性空间中的向量。换句话说，所有对象的表示

空间可以由矩阵进行数值化表示，其中每列是局部图案特征（即，所有时间点属于唯一体素），并且每行表示与单个刺激有关的响应向量。使用表征空间概念的主要优点是我们可以将机器学习中的方法应用推广到不同的测量模式。

多被试者 fMRI 分析是人脑解码中的一个具有挑战性的问题。实际上，多被试者 fMRI 图像必须在被试者之间对齐，以便考虑主体间的差异性。技术上有两种主要的校准方法，包括生理结构校准和功能校准，它们可以协同工作。然而，生理结构校准只能有限地提高准确性，因为功能区的大小，形状和解剖位置在不同被试者之间是不同的。功能对齐探索精确对齐 fMRI 图像的方法。在这里，我们尝试寻找一个共同或共享的空间，使得类内刺激之间的相关性将被最大化，并且类间神经活动相互之间具有显著的距离。在监督学习中，共享空间由训练集生成，然后将用于映射测试集中的神经活动。

多变量模式（Multivariate Pattern, MVP）分类是神经科学和计算机科学之间的结合，它可以通过应用分类方法来提取和解码大脑模式。事实上，它可以预测与不同认知状态相关的神经活动模式，也可以定义决策表面以区分不同的刺激，以解码大脑并理解它是如何工作的。此外，MVP 分类使我们能够了解大脑如何存储和处理独特的刺激。此外，MVP 分类使用机器学习算法对响应模式进行分类，将每个神经响应与实验条件相关联。模式分类涉及在神经表示空间中定义区间，区间中的所有响应向量表示相同类别的信息，例如刺激类别，有人参与的刺激或认知状态。它还可以用于寻找精神疾病的新疗法，甚至可以创建新一代的用户界面。

MVP 分类分析首先将数据划分为独立的训练和测试数据集。然后，在训练数据上提取用来确定每类神经反应向量所处区域的决策规则。不同条件下的区间之间的边界称为决策面。然后在独立测试数据上测试分类器的有效性。为了有效的泛化测试，测试数据必须独立于分类器的训练，包括数据预处理。接着，对每个测试数据响应向量根据由条件决定的区间进行分类，即所处区间的类别。

分类器准确度是正确分类的测试向量的百分比。通过检查混淆矩阵，可以对分类器性能进行更具启发性的评估。混淆矩阵表示每个实验条件下所有分类的频率，包括有关错误分类的详细信息。对错误分类的检查能够提供哪些条件最有区别或更相似的信息。可以使用表征相似性分析（Representational Similarity Analysis, RSA）中的其他方法分析这些信息。

作为 fMRI 分析的基本方法之一，RSA 评估分散注意力的认知任务之间的相似性（或距离）。RSA 检查表示空间内关于响应向量之间距离的表示结构。例如机器学习中的聚类分析，所有响应向量对之间的完整距离集称为相异度矩阵（Dissimilarity Matrix, DSM）。尽管 MVP

分类分析了不同条件下的载体是否明显不同，但 RSA 分析了它们之间的相互关系。这种方法具有几个优点。首先，RSA 可以揭示不同脑区的表征不同，即使 MVP 分类在这些区域中是等同的。其次，通过将响应向量的位置从一组特征坐标转换为向量之间的一组距离，表示空间的几何形状现在处于不依赖于特征坐标轴的格式。第三，RSA 可以比较不同物种的神经活动。RSA 技术的主要缺点是它们不能创建任何认知模型，并且必须重复整个 RSA 过程来分析新被试者。因此，RSA 技术在计算上的效率不高。

过去十五年来，人们在研究人类神经活动解码方面取得了一些重大的进展，然而，仍然有几个长期存在的挑战，其中包括多被试者数据的功能校准、有判别性的特征的选择、无监督相似性分析以及用于预测神经活动的监督模型的生成等问题。本文提出新的方法来分析不同层次和应用中的认知过程，主要贡献可概括如下:

1. fMRI 研究尤其是 MVP 分析中的一个主要挑战是使用多被试数据集。一方面，多被试分析对于验证实验结果的泛化性和有效性至关重要，另一方面，多被试 fMRI 数据分析需要在不同被试者的神经活动之间进行准确的功能和生理结构校准，以便提高最终实验结果的精度。实际上，为了考虑不同被试者之间的差异性，多被试 fMRI 数据集必须对不同的被试者进行校准。如前所述，主要有两种校准方法，即生理结构校准和功能校准，它们可以一起使用。生理结构校准是采用结构 MRI 图像（例如，基于生理结构特征）进行校准的最常用方法，比如 Talairach 对齐。然而，该方法的精度较低，因为不同被试者的脑功能区的大小、形状和生理结构各不相同。实际上，在许多 fMRI 研究中生理结构校准仅用于预处理步骤。相比之下，功能校准试图直接校准不同被试者的大脑神经响应。超校准（Hyperalignment, HA）是最著名的功能校准方法之一。在数学上超校准可以通过典型相关分析（Canonical Correlation Analysis, CCA）表示，因此，可以将用于多被试 fMRI 研究的超校准定义为多视图典型相关分析。由于采用无监督的 CCA 技术来解决超校准问题，因此可能无法针对 MVP 分析优化该解决方案。换句话说，CCA 只是找到一组映射来最大化所有被试者在相同时间点的功能活动（体素层次）之间的相关性，然而它必须最大化同类刺激（来自同一类别）之间的相关性，并且消除不同类别刺激之间的相关性。实际上，这是机器学习中的常见问题，例如，在分类分析中主要使用线性判别分析（Linear Discriminant Analysis, LDA）而不是主成分分析（Principal Component Analysis, PCA），其中线性判别分析使用诸如类标签之类的监督信息或样本之间的相似性来改进分类方法的性能。我们首先提出了一种称为局部判别超校准（Local Discriminant Hyperalignment, LDHA）的监督超校准方法，将局部判别分析（Local Discriminate Analysis, LDA）的概念融合到 CCA 中以改进超校准方法的性能。实际上，局部性的概念是基于训练集中的刺激类别（类

标签）来定义的，该方法首先为每个类别的刺激生成两个集合，即作为类内邻域的最近同类刺激集合和作为类间邻域的不同类别刺激集合，然后我们最大化类内邻域之间的相关性并且使得类间邻域之间的相关性接近于零，以此来产生更好的超校准的解。作为一种无监督的可行方法，我们还提出了一种新的核方法，称为深度超校准（Deep Hyperalignment, DHA）。DHA 采用深度核函数，包括一个多层神经网络，它可以实现任意非线性函数，而且它使用秩为 m 的奇异值分解 (rank-m Singular Value Decomposition, SVD) 和随机梯度下降 (Stochastic Gradient Descent, SGD) 进行优化。当把训练好的 DHA 模型运用到新的被试者数据上时，不需要使用训练数据。因此，DHA 在大型数据集上的运行时间较少。此外，DHA 不受固定表示空间的限制，因为 DHA 中的核是多层神经网络，它可以分别为每个被试者模拟任何非线性函数来将大脑活动映射到一个公共空间。DHA 与正则化超校准（Regularized Hyperalignment, RHA）和多视图潜在语义分析（Multiview Latent Semantic Analysis, MLSA）有关，实际上，DHA 与上述方法之间的主要区别在于深度核函数。核超校准（Kernel Hyperalignment, KHA）等同于 DHA，DHA 中所提出的深度网络被用作核函数。DHA 可以看作是使用随机梯度进行优化的多试图正则化深度典型相关分析（Deep Canonical Correlation Analysis, DCCA）。另外，当深度广义典型相关分析（Deep Generalized Canonical Correlation Analysis, DGCCA）通过使用正则化和秩为 m 的 SVD 为功能校准进行重新公式化表示时，DHA 与 DGCCA 有关。最后，我们发现在二分类问题上 DHA 具有较好的性能，然而在多分类问题上 LDHA 的性能更好。

2. MVP 分类分析首先将数据划分为两个各自独立的训练和测试数据集。然后，在训练数据上开发决策规则，该决策规则可以确定每类神经响应向量的范围。不同条件的扇区之间的边界称为决策面。然后在独立测试数据上测试分类器的有效性。对于有效性的泛化测试，测试数据不能出现在整个分类器训练过程中，包括数据预处理阶段。然后将每个测试数据响应向量分类为与其所在扇区相关的条件的另一个示例。从技术上讲，MVP 分类确实是一个具有挑战性的问题。首先，大多数 fMRI 数据集是有噪声并且稀疏的，这会降低 MVP 方法的性能。第二个挑战是如何定义感兴趣区域。如前所述，功能磁共振成像技术使我们能够研究不同脑区的信息。因此，了解不同刺激对大脑区域的影响是非常重要的，特别是在复杂的任务中（同时做一些简单的任务，如在观看照片的同时敲击按钮）。一方面，大多数之前的研究手动选择了 ROI，另一方面，定义错误的 ROI 会明显降低 MVP 方法的性能。另一个挑战是研究大脑的成本，把属于同一类别的不同数据集组合起来，可以视为该问题的解决方案，但是数据必须在标准空间中进行归一化，归一化的过程会增加时间和空间的复杂性并降低 MVP 技术的鲁棒性，尤其是在基于体素级别的方法中。最后一个挑战是结果的可视化，作为机器学习技术，MVP

代表体素或网络连接等数据的数值结果，有时，神经科学家很难找到生成结果与认知状态之间的关系。由于大多数 fMRI 数据集是高维，有噪声且稀疏的，因此一些研究采用了特征选择或提取方法。研究者可以在两个级别中选择神经活动，即体素尺度或时间尺度。在体素尺度的研究中，一些技术将原始体素的特征投射到嵌入空间，其中映射的特征既可以含有丰富的信息，又能保证其线性性质。基于组件的方法，例如 PCA，LDA 或独立成分分析（Independent Component Analysis, ICA）是体素尺度最流行的技术。作为另一种选择，SearchLight 方法通过分析神经影像数据的直方图来选择特征。实际上，有两类基于 SearchLight 的方法，即基于信息的技术和基于空间分辨率的方法。基于信息的方法仅寻找具有最高强度的体素，但基于空间分辨率的技术则考虑了所选体素的位置。实际上，本论文还提出了一种基于人脑生理结构结构来选择特征的新方法。通过使用基于组件的方法来选择来自任何脑区（相关或不相关）的神经活动，而基于生理结构的方法则是使用空间信息来选择特征。基于时间尺度的方法侧重于选择时间点的子集而不是使用整个时间序列。先驱们的研究采用基于组件的方法（如 PCA 和 ICA）来选择时间点的子集，而我们提出了一种新的特征分析技术。我们所提出方法的主要思想是非常简明易懂的。当使用氧气的水平最大化时，所提出的方法不是分析整个时间序列，而是估计每个刺激的脑影像的快照。结果表明，该方法可以自动减少脑影像的稀疏性。这一方法的过程分为三个阶段：首先，通过在设计矩阵的平滑版本中找到局部最大值来选择脑影像的快照。然后，以三个步骤生成特征，包括归一化到标准空间，以自动检测的生理结构区域的形式分割快照，以及通过 ROI 水平中的高斯平滑去除噪声。最后，我们提出了两种学习方法来分析提取的特征，即无监督学习和监督学习。本文提出了一种聚类集成方法以应用无监督学习，其中神经活动之间的相似性或距离可以跨被试者进行比较。作为监督学习的替代方案，我们在二元 $\ell1$ 正则化 SVM 分类器上应用集成分类（即，bagging 技术），其中这些分类器通过在生理结构区域的水平中使用每个神经活动来创建，即，每个快照表示在一个独特刺激下的神经活动。

3. 作为 fMRI 分析的基本方法之一，表征相似性分析是一种监督方法，用于评估不同的认知任务之间的相似性（或距离）。在实践中，RSA 可以在数学上被看作为多组回归问题，即用于在神经活动矩阵和设计矩阵之间进行映射的线性模型。经典 RSA 采用基础的线性方法，例如普通最小二乘法（Ordinary Least Squares, OLS）或一般线性模型（General Linear Model, GLM）。实际上，这些方法不能在真实世界数据集上提供比较好的性能，例如具有广泛感兴趣区域或全脑 fMRI 数据的数据集。一方面，大多数 fMRI 数据集中的体素数量多于时间点数量。因此，神经活动的矩阵可能不是满秩的。另一方面，所提到的方法必须计算神经活动的协

方差矩阵的逆，以解决 RSA 问题。当协方差矩阵包括低信噪比（Signal-to-Noise Ratio, SNR）时，该逆降低了结果的稳定性。作为第一组常用方法，一些新的 RSA 方法使用贝叶斯技术。作为这些算法之一，贝叶斯 RSA（Bayesian Representational Similarity Analysis, BRSA）将协方差矩阵视为超参数生成模型，然后根据神经活动计算该矩阵。尽管贝叶斯方法可以显著改善 SNR 问题甚至处理一些非线性数据集，但它们仅限于受限制的变换函数（超参数的高斯分布）。作为经典方法中的另一个问题，OLS 和 GLM 没有使用正则化项来避免过拟合。第二组常用方法侧重于正则化问题。虽然岭回归方法利用额外的范数 $\ell2$ 来解决上述问题，但最小绝对收敛选择算子（Least Absolute Shrinkage and Selection Operator, LASSO）方法使用 $\ell1$ 范数来规范回归问题。作为另一种选择，Elastic Net 方法在 $\ell1$ 和 $\ell2$ 范数之间进行权衡。此外，其他技术开发了新的正则化项，例如八边形收缩和回归聚类算法（Octagonal Shrinkage and Clustering Algorithm for Regression, OSCAR）或有序加权 $\ell1$（Ordered Weighted $\ell1$, OWL）。作为主要问题，这些方法总是认为特征之间的关系是线性的。我们开发了深度表征相似性分析（Deep Representational Similarity Analysis, DRSA）用于相似性分析。DRSA 采用深度核函数，将非线性神经活动转换为线性嵌入空间，然后评估该空间中映射特征之间的相似性（或距离）。此外，它采用了一个新的正则化项，可以在不同类别的刺激的相关性和协方差之间进行权衡。由于 DRSA 采用基于梯度的优化方法，因此评估高维神经影像数据（例如全脑图像）可以有效节省时间。

4. 为了解码人脑中的神经活动，多体素模式分析（Multi-Voxel Pattern Analysis, MVPA）技术将机器学习方法应用于基于任务的 fMRI 数据集。MVPA 的主要问题之一是不平衡分类问题。在实践中，有两种方法来处理这个问题，即设计不平衡分类器，或将不平衡问题转换成一组平衡分类模型。以前的研究表明，不平衡分类器的性能可能不稳定，尤其是当我们的数据集中存在稀疏性和噪声时。由于 fMRI 数据集主要包括噪声和稀疏性，本文选择了集成方法。从技术上讲，集成学习还包含两组解决方案，即 bagging 或 boosting。其中 bagging 同时生成所有分类器，然后将它们全部组合为最终模型，然而 boosting 逐渐创建每个分类器，以便通过跟踪先前迭代的错误来提高每次迭代的性能。我们只需要注意，集成学习可以用于平衡和不平衡问题。事实上，主要的区别来自采样策略。在平衡问题中，采样方法应用于整个数据集，而大类的实例则在不平衡问题中进行采样。本文提出了 AdaBoost 算法的一个新分支，称为不平衡 AdaBoost 二分类（Imbalance AdaBoost Binary Classification, IABC），以显著提高 fMRI 分析中最终模型的性能。简而言之，该算法首先将不平衡 MVPA 问题转换为一组平衡问题，然后，它迭代地将决策树应用于每一个平衡问题中，最后，使用 AdaBoost 来生成最终模型。在

所提出的方法中，基于先前迭代的误差（失败的预测）生成最终组合的每个分类器（树）的权重，以逐渐改善最终模型的性能。在本文中，多分类器是一种预测模型，用于将提取的特征映射到刺激类别。通常，有两种应用多分类的技术。第一种方法直接创建分类模型，如多类支持向量机或神经网络。相反，分解设计（间接）使用二分类器数组来解决多类问题。本文利用纠错输出码（Error-Correcting Output Codes, ECOC）作为间接多类方法，以扩展所提出的二分类器，用于多类预测。我们的 ECOC 包括三个组件，即基本算法，编码矩阵和解码过程。由于本文使用一对所有编码策略，所以提出的二分类器被用作 ECOC 中的基础算法，其中它为每类刺激生成二分类器。换句话说，将每个独立的刺激类别与其他类别进行比较。因此，此策略中的分类器数量与类别数量完全相同。在解码阶段，通过将脑响应应用于基础算法而生成的二分类预测被分配给具有最接近 Hamming 距离的编码矩阵中的类别。

对 20 种不同的真实神经影像数据集进行的实验研究证实，所提出的方法的性能优于其它经典和前沿算法。除了论文中的理论和实证研究之外，我们还创建了一个基于图形用户界面 (Graphical User Interface, GUI) 的工具箱使得研究易于重现并向公众开放，该工具箱可以运行包括本文提出的方法在内的基于任务的 fMRI 图像分析方法的标准流程，目前该工具箱已在 `https://easyfmri.github.io` 上公开。此外，我们还提供了一个数据仓库用于共享本文所使用的数据集，该仓库位于 `https://easyfmridata.github.io`。