Towards Robust Machine Learning Models for Data Scarcity

by

Jie Zhang

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved March 2020 by the
Graduate Supervisory Committee:

Yalin Wang, Chair
Huan Liu
Cynthia Stonnington
Jianming Liang
Yezhou Yang

ARIZONA STATE UNIVERSITY

May 2020

ABSTRACT

Recently, a well-designed and well-trained neural network can yield state-of-the-art results across many domains, including data mining, computer vision, and medical image analysis. But progress has been limited for tasks where labels are difficult or impossible to obtain. This reliance on exhaustive labeling is a critical limitation in the rapid deployment of neural networks. Besides, the current research scales poorly to a large number of unseen concepts and is passively spoon-fed with data and supervision.

To overcome the above data scarcity and generalization issues, in my dissertation, I first propose two unsupervised conventional machine learning algorithms, hyperbolic stochastic coding, and multi-resemble multi-target low-rank coding, to solve the incomplete data and missing label problem. I further introduce a deep multi-domain adaptation network to leverage the power of deep learning by transferring the rich knowledge from a large-amount labeled source dataset. I also invent a novel time-sequence dynamically hierarchical network that adaptively simplifies the network to cope with the scarce data.

To learn a large number of unseen concepts, lifelong machine learning enjoys many advantages, including abstracting knowledge from prior learning and using the experience to help future learning, regardless of how much data is currently available. Incorporating this capability and making it versatile, I propose deep multi-task weight consolidation to accumulate knowledge continuously and significantly reduce data requirements in a variety of domains. Inspired by the recent breakthroughs in automatically learning suitable neural network architectures (AutoML), I develop a nonexpansive AutoML framework to train an online model without the abundance of labeled data. This work automatically expands the network to increase model capability when necessary, then compresses the model to maintain the model efficiency.

In my current ongoing work, I propose an alternative method of supervised learning that does not require direct labels. This could utilizes various supervision from an image/object as a target value for supervising the target tasks without labels, and it turns out to be surprisingly effective. The proposed method only requires few-shot labeled data to train, and can self-supervised learn the information it needs and generalize to datasets not seen during training.

DEDICATION

To my family,

for their everlasting love and supports.

To my friends,

for their encouragement and company.

Along this journey.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Dr. Yalin Wang, for his guidance, encouragement, and support during my dissertation research. He is an outstanding mentor, an easygoing friend, and the most dedicated researcher I have ever known. The experiences with him are my lifelong assets. I would like to thank my dissertation committee members, Dr. Huan Liu, Dr. Cynthia Stonnington, Dr. Jianming Liang, and Dr. Yezhou Yang, for their valuable interactions and feedback. I also thank Dr. Paul M. Thompson from University of South California, Dr. Kewei Chen of Banner Alzheimers Institute and Banner Good Samaritan PET center, Dr. Ye from University of Michigan for instructions and suggestions on experiments and paper writings. Thanks to my academic advisor Arzuhan Kavak for her always support.

I was lucky to work as interns in Samsung Research America and Apple Inc. with amazing colleagues and mentors: Dr. Xiaolong Wang, Jingwen Zhu, Dr. Yang Song, Dr. Heming Zhang, Boyu Wang, Dr. Dawei Li, Kai Xu, Dr. Shalini Ghosh, Junting Zhang from Samsung Research America and Dr. Vinay Sharma, Dr. Abhishek Singh from Apple Inc. Because of you, my life became much easier in new environments; because of you, I enjoyed four wonderful and productive internship; and because of you, I was able to contribute my knowledge to exciting projects. Thank you for everything.

Members of our Geometry Systems Lab inspired me a lot through discussions, seminars, and project collaborations, and I would like to thank the following people for their valuable interactions and encouragement from them during my most suffer time in the Ph.D. study: Especially thanks to Duyan Ta, for his always take care, encourage, company, help, supports and valuable suggestions for my every presentations, and Thanks to Dr. Qunxi Dong, Dr. Jie Shi, Wen Zhang, Liang Mi, Yonghui Fan, Jianfeng Wu, Yanshuai Tu, Yanxi Chen, Mohammad Farazi, Nahid Ul Islam.

Finally, I am deeply indebted to my dear mother and father for their love and strong support during my graduate study. This dissertation is dedicated to them. It is also a memorable time for the years I spend at ASU due to many other friends, although we are from different departments, they made my life becomes colorful and artistic. Last, thanks to Dr. Qingyang Li for his endless support and encouragement during this journey.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Chapter 1

INTRODUCTION

Recently, a well-designed and well-trained neural network can yield state-of-the-art results across many domains, including data mining, computer vision, and medical image analysis. But progress has been limited for tasks where labels are difficult or impossible to obtain. This reliance on exhaustive labeling is a critical limitation in the rapid deployment of neural networks. Besides, the current research scales poorly to a large number of unseen concepts and is passively spoon-fed with data and supervision.

The first and the most challenging problem in this dissertation is the source and quality of the training data are limited, so-called "data scarcity". This often happens in medical image analysis, and it is also hard to get large-scale data sets or sufficient data for building an excellent deep learning model. To address the above problem, I first propose a Hyperbolic space Sparse Coding (HSC) framework (Zhang *et al.*, 2016a), in which the Farthest point sampling with Breadth-first Search (FBS) algorithm is proposed to construct ring-shaped feature patches from hyperbolic space and patch-based hyperbolic sparse coding algorithm is developed to reduce the data dimensionality while only a small number of samples are available. In this regard, machine learning has been playing a pivotal role to overcome this so-called "large p, small n" problem (Li *et al.*, 2016a,b; Zhu *et al.*, 2017).

Sparse Coding (SC) (Lee *et al.*, 2006) has been proposed to use a small number of basis vectors to represent local features effectively and concisely and help image content analysis. However, most existing SC works focused on the prediction of the target at a single time point Mairal *et al.* (2009) or as a single-task problem Zhang *et al.* (2016a) or single region-of interest Zhang *et al.* (2016a,c, 2017a). In general, a

joint analysis of tasks from multiple sources is expected to improve the performance but remains a challenging problem. Multi-Task Learning (MTL) has been successfully explored for regression with different tasks. The idea of MTL is to utilize the intrinsic relationships among multiple related tasks in order to improve the prediction performance. One way of modeling a multi-task relationship is to assume all tasks are related, and the task models are connected to each other (Evgeniou *et al.*, 2005), or the tasks are clustered into groups (Zhou *et al.*, 2012). Alternatively, one can assume that tasks share a common subspace (Chen *et al.*, 2009), or a common set of features (Argyriou *et al.*, 2008). To this end, I proposed an unsupervised multi-task sparse coding algorithm termed Multi-Resemblance Multi-Target Low-Rank Coding (MMLC) (Zhang *et al.*, 2017c), to learn the different tasks simultaneously which utilizes shared and individual dictionaries to encode both consistent and individual imaging features for multi-task learning and longitudinal image data analysis.

Deep learning algorithms simulate the hierarchical structure of the human brain, process data from lower levels to higher levels, and gradually compose more and more semantic concepts. Deep learning also requires a massive amount of training dataset as classification accuracy and the generalization ability of a deep neural network mainly depends on the quality and the size of the dataset. However, insufficient dataset is one of the most significant barriers to the success of deep learning in medical image analysis and many other applications. Therefore, I propose a deep domain adaptation algorithm termed Deep natural-Domain Multi-ROIs learning (DDAML) (Zhang *et al.*, 2017d) to leverage the rich knowledge from a large-amount labeled natural dataset and adapt on the limited amount labeled brain image data.

Although a general unsupervised SC may overcome the missing label problem to obtain the sparse features, there is still a need that considers leveraging the labeled data with the consistent time series features to learn a more strong sparsity pat-

tern. To comprehensively capture temporal-subject sparse features, I invent a supervised time-sequence dynamically hierarchical network termed Temporally Adaptive-Dynamic Sparse Network (TaDsNet) (Zhang and Wang, 2020) to uncover the sequential correlation with only small amount subject-level image. It guarantees high predictive power by dynamically mining the labeled data of the projection dictionary matrix within the network hidden layer and adaptively changing the sparsity of the network across the hierarchical layers.

The above four works either predict the target value as the isolated single task learning problem (Zhang *et al.*, 2016a) or develop joint analysis schemes as the multi-task learning problem (Zhang *et al.*, 2017c,d; Zhang and Wang, 2020). These algorithms do not take into account that the real-world data are obtained in a continuous sequence rather than a uniform batch. Different batches of data arrive periodically (e.g., monthly, seasonally, or yearly) with the data distribution changing over time. This presents an opportunity for lifelong learning, whose primary goal is to learn consecutive tasks without forgetting the knowledge acquired in the past (e.g. with less longitudinal data) and leverage the previous knowledge to build a lifelong learning machine to achieve general artificial intelligence. One simple way is to fine-tune the model for every new data set; however, the retrained representations may adversely affect the old tasks, causing them to drift from their optimal solution. This way can cause "catastrophic forgetting", a phenomenon where training a model with new tasks interferes with the previously learned old knowledge, leading to performance degradation or even overwriting of the old knowledge by the new ones. To overcome the above "catastrophic forgetting" problem, many approaches have been proposed (Kirkpatrick *et al.*, 2017; Li and Hoiem, 2017; Lopez-Paz *et al.*, 2017). Incorporating this capability and making it versatile, holistic and intelligent, I propose a Deep Multi-order Preserving Weight Consolidation (dMopWC) (Zhang and Wang,

2019a) to continually learn the time-order of sequence data without losing statistical power on less longitudinal data and ensure that the old and new tasks correlation is respected.

Inspired by the recent breakthroughs in automatically learning suitable neural network architectures (AutoML), I further develop a nonexpansive AutoML framework for continual learning termed regularized, expand and compress (REC) to train an online model without the abundance of labeled data. This work automatically expands the network to increase model capability for unseen classes and smart compress the expanded model to a suitable size in order to maintain the model efficiency.

Last, I propose an alternative method of supervised learning that does not require direct labels. The intuition is that we might obtain various properties or supervision from an image/object without the label. Therefore, we could utilize these properties as a target value for supervising the target tasks. We observe that this kind of "self-supervision" on how the output behaves rather than what is it, and it turns out to be surprisingly effective in learning a variety of vision tasks. My current ongoing work presents an original approach for self-supervised learning features by using outside supervision rather than direct labels. We argue that the proposed method only requires few-shot labeled data to train, and it can act as supervised learning the information it needs, but use as same as unsupervised learning information. Therefore, the proposed algorithm can generalize to datasets not seen during training.

The remainder of this dissertation is organized as follows. In Chapter 2, I detail the HSC and show that the HSC is convergent and enjoys strong theoretical guarantees. In Chapter 3, I introduce MMLC and the updating rules of dictionaries and sparse codes. In Chapter 4, I explain the DDAML and summarize TaDsNet in Chapter5. Later, I present the continual learning works dMopWC and REC in Chapter 6 and Chapter 7, respectively. Finally, I conclude the dissertation and briefly descript my

ongoing work as well as point out broader impacts and promising future research directions in Chapter 8.

Chapter 2

HYPERBOLIC STOCHASTIC CODING WITH RING-SHAPED PATCH SELECTION

2.1 Introduction

Alzheimer's Disease (AD), an irreversible neurological degeneration, is the most common disease in older adults. It is generally agreed that accurate presymptomatic diagnosis and preventive treatment of AD could have enormous public health benefits. Brain structural magnetic resonance imaging (sMRI) analysis has the potential to provide valid diagnostic biomarkers of the preclinical stage as well as symptomatic AD. Prior work has demonstrated that surface-based analyses (Thompson *et al.*, 2000; Fischl, 2012) can offer advantages over volume measures, due to their sub-voxel accuracy and the capability of detecting subtle subregional changes. Recently, brain surface morphometric maps have been integrated with machine learning algorithms to classify individual subjects into different diagnostic groups (Sun *et al.*, 2009; Ferrarini *et al.*, 2008a; Wang *et al.*, 2013), which offers a promising approach to computer-aided diagnosis and prognosis by leveraging both sensitive surface-based brain image features and powerful machine learning techniques.

In brain imaging research, a practical approach to model brain landmark curves is to model them as surface boundaries by cutting open cortical surfaces along these landmarks. Thus they are modeled as open boundaries to be matched across subjects (Shi and Wang, 2019; Tsui *et al.*, 2013) or be used as shape indices (Shi *et al.*, 2017; Zeng *et al.*, 2013). Similarly, adding open boundaries have been proved to be useful in modeling ventricular surfaces which have a concave shape and complex branching

topology (Wang *et al.*, 2010; Shi *et al.*, 2015). We call these genus-zero surfaces with more than two open boundaries as *general topological surfaces* and hyperbolic geometry has been demonstrated to be useful to model general topological surfaces. However, most of prior hyperbolic space-based brain imaging methods have been focused on studying difference between diagnostic groups. To develop brain imaging methods for personal medicine research, it would be advantageous to design powerful machine learning methods that work on general topological surfaces for early AD diagnosis and prognosis on an individual basis.

There are at least two challenges to directly apply vertex-wise surface features to the classification research. The first is the strong local feature variance on the measured surface statistics and the second is the so-called *high dimension-small sample problem*. To address these two problems, we first adopt patch-based local image analyses (Mairal *et al.*, 2008) to improve signal-to-noise ratio (SNR) in the surface features. Following that, we propose a novel hyperbolic sparse coding algorithm, termed hyperbolic stochastic coding (HSC), to extract critical low-dimensional shape features from the hyperbolic surface maps. Compared with the traditional online dictionary learning (ODL) work (Mairal *et al.*, 2009), HSC dramatically improves computation efficiency while enjoying strong theoretical guarantees.

Although HSC can extract critical low-dimensional shape features, the hyperbolic space is different from the original Euclidean space, because the structure is more complicated and demands more efforts for selecting patches based on its topological structure. The common rectangle patch construction cannot be directly applied to the hyperbolic space. We thus invent a farthest point sampling with breadth-first search (FBS) to obtain ring-shaped patches for sparse coding initialization. In our prior work (Zhang *et al.*, 2016a), we introduced hyperbolic space sparse coding with some simple illustrative examples. In the present work, we provide a detailed and complete

description of hyperbolic space sparse coding algorithm and provide a complete theoretical analysis of the hyperbolic space sparse coding convergence. Moreover, here we carefully explore a few more applications with our hyperbolic space sparse coding framework on Alzheimers Disease Neuroimaging Initiative (ADNI) dataset (Weiner *et al.*, 2012) and the results demonstrate the potential of our work for these applications.

We summarize our HSC contributions into threefold as follows. First, we propose an efficient hyperbolic space sparse coding algorithm – HSC. To the best of our knowledge, HSC is the first sparse coding framework which is designed for general topological surfaces admitting the hyperbolic geometry. Second, in order to better initialize the dictionary for sparse coding on the hyperbolic parameter domain, we propose a ring-shaped patch selection algorithm – FBS – to capture the surface features. The extracted patch structure help reduce feature noises and enhance statistical power of the computed surface TBM features. Third, the HSC is theoretically rigorous and computationally efficient, which is more than 30 times faster than traditional online dictionary learning. This is the first time that we give a theoretical convergence analysis of the proposed HSC algorithm and the same analysis framework may be generalized to prove the convergence of a related work – sparse stochastic coding (SSC) work (Lin *et al.*, 2014). We validate our proposed HSC and FBS on two datasets and the experimental results demonstrate that the proposed algorithms outperform some other work on both running time and classification accuracy.

2.2 Related Work

2.2.1 Brain Morphometry Study

Deformation-based morphometry and tensor-based morphometry are well studied in the analysis of brain imaging in structure volumes and shapes. Deformation-based morphometry (DBM) (Ashburner *et al.*, 1998; Chung *et al.*, 2001; Wang *et al.*, 2003; Chung *et al.*, 2003) uses deformations obtained from the nonlinear registration of brain images to a common anatomical template, to infer 3D patterns of statistical differences in brain volume or shape. Tensor-based morphometry (TBM) (Thompson *et al.*, 2000; Chung *et al.*, 2008a) is a related method, which examines spatial derivatives of the deformation maps registering brains to a common template. Morphological tensor maps are used to derive local measures of shape characteristics such as the Jacobian determinant, torsion or vorticity. DBM, by contrast, analyzes 3D displacement vector fields encoding relative positional differences in anatomical structures across subjects, after mapping all brain images to a common stereotaxic space (Thompson *et al.*, 1997; Cao *et al.*, 1997). One advantage of TBM for studying brain structure is that it also derives local derivatives and tensors from the deformation for further analysis. When applied to surface models, surface multivariate TBM (mTBM) (Wang *et al.*, 2010) may make use of the Riemannian surface metric to characterize the directions of local surface abnormalities and further improve the statistical power in surface-based brain image analyses.

2.2.2 Sparse Coding

Existing feature dimension reduction approaches include feature selection (Fan *et al.*, 2005), feature extraction (Saadi *et al.*, 2007) and sparse coding-based methods (Vounou *et al.*, 2010). In most cases, information is lost when mapping into

a lower-dimensional space. By defining a better lower-dimensional subspace, sparse coding (Lee *et al.*, 2006; Mairal *et al.*, 2009) may limit such information loss. It has been previously proposed to learn an over-complete set of basis vectors (dictionary) to represent input vectors efficiently and concisely (Donoho and Elad, 2003). Sparse coding has shown to be efficient for many tasks such as image deblurring(Yin *et al.*, 2008), super-resolution (Yang *et al.*, 2010), classification (Mairal *et al.*, 2009), functional connectivity (Zhang *et al.*, 2018b; Lv *et al.*, 2015b, 2017; Jiang *et al.*, 2015a; Lv *et al.*, 2015a), and structural morphometry analysis (Zhang *et al.*, 2017c; Li *et al.*, 2017).

2.3 Methods

The major computational steps of our proposed work are illustrated in Fig. 2.1 where we take a left ventricular surface as an example. The new framework can be divided into two stages. In the first stage, we perform ventricular surface reconstruction from MRI data, surface registration and surface TBM feature computation. In the second stage, we build ring-shaped patches on the hyperbolic parameter space by FBS to initialize the original dictionary. Hyperbolic stochastic coding and max-pooling are performed for dimension reduction. Following that, Adaboost is adopted to diagnose different clinical groups and predict future AD conversions. The pipeline source code is publicly available at `http://gsl.lab.asu.edu/software/pass-mp/`.

2.3.1 Brain Surface Registration with Hyperbolic Ricci Flow and Harmonic Map

Taking a left ventricular surface S as an example, the corresponding framework is summarized in Algorithm 1 and Fig. 2.1 (c). Its critical steps are shown in Fig. 2.2.

Following our prior work (Shi *et al.*, 2015), three horns of a ventricular surface are identified and three cuts $\{\gamma_1, \gamma_2, \gamma_3\}$ are made on these horns (Fig.2.2 (a)). The

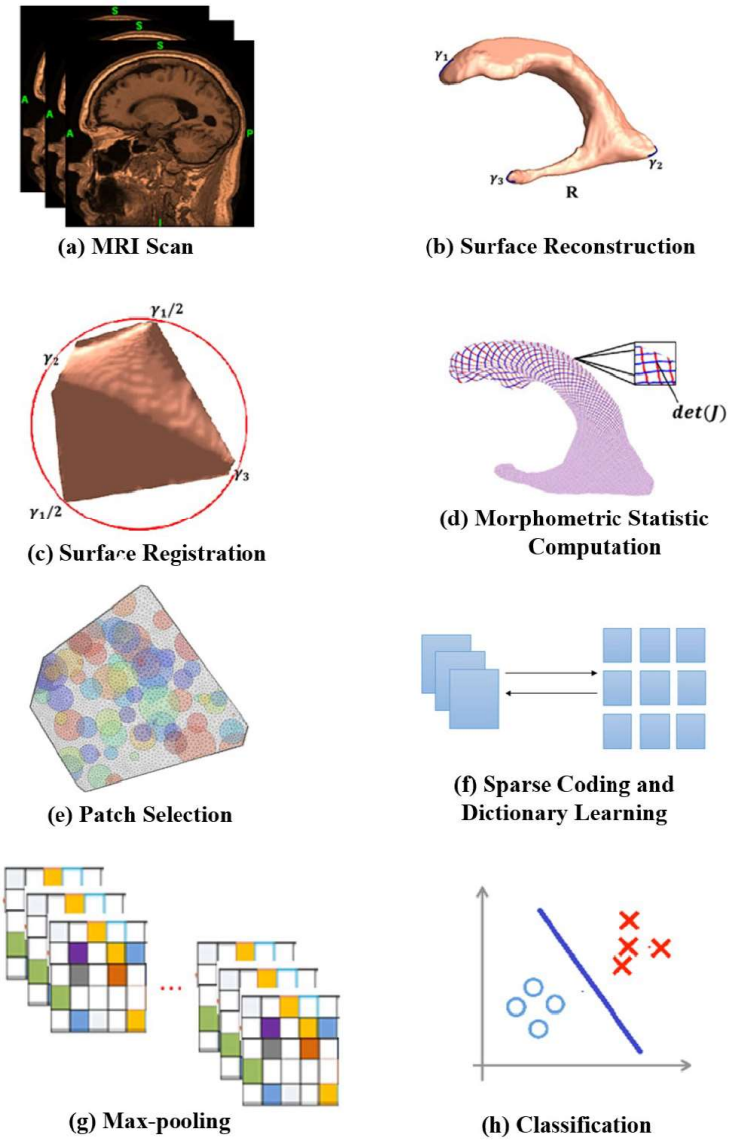


Figure 2.1: The Major Processing Steps in the Proposed HSC Framework.

locations of the cuts are motivated by examining the ventricular topology and kept consistent across subjects (Wang *et al.*, 2010). We term this step as *topology optimization*. As a result, each ventricular surface becomes a topologically multiply connected surface and admits the hyperbolic geometry. It can be mapped to the hyperbolic space. We apply the hyperbolic Ricci flow method to compute its discrete hyperbolic uniformization metric. For more details of hyperbolic Ricci flow, please

Algorithm 1: Brain surface registration with hyperbolic Ricci flow and harmonic map

Input : Brain surface S with more than two open boundaries.

Output: Klein model of S

begin

Compute the hyperbolic uniformization metric of S with hyperbolic Ricci Flow.

Compute the fundamental group of paths on S and, together with original boundaries, obtain the simply connected domain $\bar{S}$.

Embed S onto the Poincaré disk with its hyperbolic metric and its simply connected domain $\bar{S}$, we obtain the fundamental domain of S.

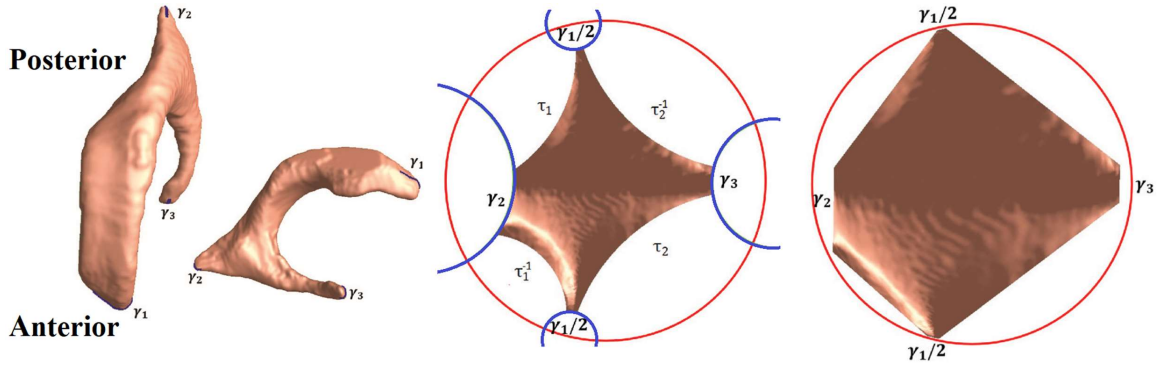
Tile the fundamental domain of S with its Fuchsian group of transformations to get a finite portion of the universal covering space of S.

Compute the positions of the paths in the fundamental group as geodesics in the universal covering space. By slicing the universal covering space along the geodesics, we obtain the canonical fundamental domain of S.

Convert the canonical Poincaré disk to the Klein model and construct the harmonic map between S and a selected template surface.

refer to (Shi *et al.*, 2015).

With the hyperbolic uniformization metric, we can embed S onto the Poincaré disk. The simply connected domain of S should be obtained by computing its fundamental group for 2D embedding. This work computes the fundamental group of a multiply connected surface by choosing the longest boundary on it and tracing a path from that boundary to one of the endpoints of every other boundary. The paths are traced with Dijkstra's algorithm avoiding collisions (Li *et al.*, 2009). As shown



(a) Topology optimization of a left ventricular surface

(b) Parametric domain of left ventricle in (a)

(c) Klein model of left ventricle in (a)

Figure 2.2: Modeling Ventricular Surface with Hyperbolic Geometry. (a) Shows Three Identified Open Boundaries, γ_1, γ_2, γ_3, on the Ends of Three Horns. After that, Ventricular Surfaces can be Conformally Mapped to the Hyperbolic Space. (b) and (c) Show the Hyperbolic Parameter Space, Where (b) is the Poincaré Disk Model and (c) is the Klein Model.

in Fig. 2.2 (b), $\{\gamma_1, \gamma_2, \gamma_3\}$ are some consistent anchor curves automatically located on the end points of each horn. On the parameter domain, τ_1 is an arc on the circle which passed on endpoint of $\frac{\gamma_1}{2}$ and one endpoint of γ_2, and is orthogonal to $|z| = 1$. To guarantee the consistency of the geodesic curve computation, endpoints of γ_1, γ_2 and γ_3 have to be consistent, while the initial paths τ_1 and τ_2 between them may be inconsistent (Shi *et al.*, 2015).

Given the Riemannian metric of the Poincaré disk model, the conformal factor near boundaries of the Poincaré disk embedding goes to infinity (Li *et al.*, 2009). This may introduce instability in the following computations, especially for complicated surfaces as those of human cortices. To address this problem, in the embedding algorithm, we pick the seed face to be a triangle that is close to the center of the fundamental domain of each surface. As a result, the embedding is close to the center of the Poincaré disk.

In the fundamental domain of S, the initial paths τ_1, τ_2 may be inconsistent so the initial fundamental domain of a multiply connected surface cannot serve as the

canonical parameter space for surface registration. We solve this problem by applying a *geodesic curve lifting* step to achieve consistent boundaries with the Fuchsian group of S (Shi *et al.*, 2015). A finite portion of the universal covering space, i.e., the entire Poincaré disk, can be tiled by mapping a fundamental domain to other periods with the Fuchsian transformations and gluing the transformed fundamental domains with the original fundamental domain.

In the universal covering space, we recompute the geodesics, which are the hyperbolic lines that are perpendicular to the unit circle and cross certain points in the Poincaré disk. Similar to our prior work (Shi *et al.*, 2015), we enforce them to cross the endpoints of existing boundaries. These geodesics are unique and consistent across subjects (Shi *et al.*, 2015). By slicing the universal covering space along the new geodesics, we obtain the *canonical fundamental domain* of the multiply connected surface S. To ensure the stability of geodesic computation near the boundaries, we only tile a finite portion of the Poincaré disk by gluing each undetermined boundary with a transformed fundamental domain. When lifted to 3D, the positions are also consistent across subjects.

In the canonical fundamental domain of S, all boundary curves become geodesics. As the geodesics are unique, they are also consistent when we map them back to the surface in $\mathbb{R}^3$. Furthermore, we convert the Poincaré model to the Klein model with the complex function: $z = 2z/1 + \overline{z}z$ (Shi *et al.*, 2015). It converts the canonical fundamental domains of the ventricular surfaces to a Euclidean octagon, as shown in Fig. 2.2 (c). Then we compute surface harmonic map with the Klein disk as the canonical parameter space for the following surface morphometry analysis (Shi *et al.*, 2015).

2.3.2 Surface Tensor-Based Morphometry

Suppose $\phi = S_1 \to S_2$ is a map from surface S_1 to surface S_2. The derivative map of ϕ is the linear map between the tangent spaces $d\phi : TM(p) \to TM(\phi(p))$, induced by the map ϕ, which also defines the Jacobian matrix of ϕ. The derivative map $d\phi$ is approximated by the linear map from one face $[v_1, v_2, v_3]$ to another one $[w_1, w_2, w_3]$. First, we isometrically embed the triangles $[v_1, v_2, v_3]$ and $[w_1, w_2, w_3]$ onto the Klein disk, the planar coordinates of the vertices are denoted by $v_i, w_i, i = 1, 2, 3$, which represent the 3D position of points $v_i, w_i, i = 1, 2, 3$. Then, the Jacobian matrix for the derivative map $d\phi$ can be computed as $J = d\phi = [w_3 - w_1, w_2 - w_1][v_3 - v_1, v_2 - v_1]^{-1}$.

Based on the derivative map J, the surface TBM is defined as $\sqrt{det(J)}$, which measures the amount of local area changes in a surface with the map ϕ (Fig. 2.1 (d)). As pointed out in (Chung *et al.*, 2005), each step in the processing pipeline including MRI acquisition, surface registration, etc., are expected to introduce noise in the deformation measurement. To account for the noise effects, we apply surface heat kernel smoothing algorithm proposed in (Chung *et al.*, 2005) to improve SNR in the TBM features and boost the sensitivity of statistical analysis.

2.3.3 Ring-Shaped Patch Selection

The hyperbolic space is different from the original Euclidean space. The common rectangle patch construction developed in Euclidean space (Zhang *et al.*, 2017c) cannot be directly applied to the hyperbolic space. Therefore, we proposed FBS on hyperbolic space to initialize dictionaries for sparse coding (Fig. 2.1 (e)). Fig. 2.3 (right) is the visualization of patch selection on the hyperbolic parameter domain. And Fig. 2.3 (left) projects the selected patches on the hyperbolic parameter domain back to the original ventricular surface, which still maintains the same topological

structure as the parameter domain. Different colorful patches in Fig. 2.3 represent patches covering ways on ventricle surface.

We first randomly selected a point center on the hyperbolic space, denotes by c_1, $c_1 \in V$, where V is the set of all discrete vertices on the hyperbolic space. We then find all points $c_{1,i}(i = 1, 2, ..., u)$, where u is the maximum number of connected points connecting with the ring patch center c_1 and $c_{1,i}$ is the i-th vertex from c_1. The procedure is called breadth-first search (BFS) (Patel *et al.*, 2015), which is an algorithm for searching graph data structures. It starts at the tree root and explores the neighbor nodes first, before moving to the next level neighbors. We used the same procedure to find all connected points with $c_{1,i}$, which are $c_{1,i_j}(j = 1, 2, \cdots, w_i)$. Here, w_i represents the maximum number of connected points with each specific point $c_{1,i}$. The points c_{1,i_j} are connected with $c_{1,i}$ by using the same procedure–BFS–between c_1 and $c_{1,i}$. Finally, we get a set $\mathbf{x}_1$ as follows, which is a selected patch with patch center c_1 and do not contain duplicate points. We called $\mathbf{x}_1$ is a selected ring-shaped patch on hyperbolic space.

$$\mathbf{x_1} = \{c_1, c_{1,1}, \cdots, c_{1,1_{w_1}}, \cdots, c_{1,u}, \cdots, c_{1,u_{w_u}}\}. \tag{2.1}$$

We can find all connected components of the center point c_1 which are all in set $\mathbf{x}_1$. The dimension of $\mathbf{x}_1$ is $u+w_1+\cdots+w_u = m$, we then have $\mathbf{x}_1 \in \mathbb{R}^m$. We construct the topological patches based on hyperbolic geometry and the edge connections among different points from $\mathbf{x}_1$. We use $\mathbf{x}_1$ to denote the first selected patch of the root (patch center) c_1 throughout the paper. Since we randomly select patches with different overlap degrees, we use radius $r = \max_{c_v \in V} d_V(c_v, c_1)$ to determine next patch's root c_2 position.

In this way, we can find the second patch root $c_2 \in V$ with the farthest distance r of c_1. We apply farthest point sampling (Moenning and Dodgson, 2003), because the sampling principle is based on the idea of repeatedly placing the next sample point in

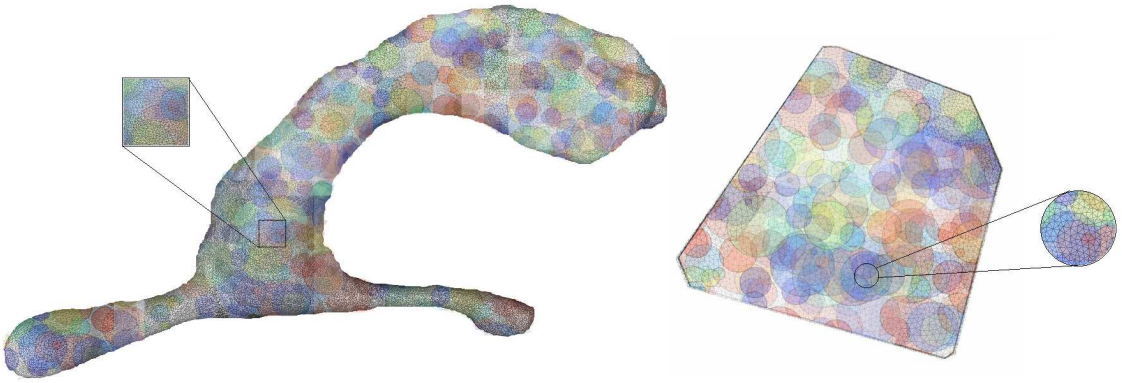


Figure 2.3: Visualization of Computed Image Patches on the Ventricle Surface (Left) and Hyperbolic Space (Right). The Zoom-in Pictures Show Some Overlapping Areas between Image Patches.

the middle of the least known area of the sampling domain, which can guarantee the randomness of the patches selection. Here, d is the hyperbolic distance in the Klein model. Given two points v' and v'', draw a straight line between them; the straight line intersects the unit circle at points a and b, so d is defined as follows:

$$d(v', v'') = \frac{1}{2}(\log \frac{|av'||bv''|}{|av'||bv''|}), \tag{2.2}$$

where $|av'| > |av''|$ and $|bv'| > |bv''|$.

Then, we can calculate:

$$c_2 = \arg\max_{c_v \in V} d_V(c_v, V_r), \tag{2.3}$$

where V_r denotes the set of selected patch centers ($V_r = \{c_1\}$ when compute c_2). Then, we add c_2 into V_r and iterate the patch selection procedure for $n = 2000$ times to get 2000 patches, which cover all vertexes according to our experience. The details of FBS are summarized in Algorithm 2.

2.3.4 Hyperbolic Stochastic Coding

We model surface TBM features as a sparse linear combination of atoms selected from a dictionary which is initialized by FBS on the hyperbolic parameter space.

Algorithm 2: Farthest point sampling with Breadth-first Search (FBS)

Input : Hyperbolic parameter space.

Output: A collection of different amount overlapped patches on topological structure.

begin

  Start with $V_r = \{c_1\}$, V denotes all discrete vertices on the hyperbolic space and V_r denotes the set of selected patch centers.

  for *T=1 to n* **do**

    for *r determine sampling radius* **do**

      Find all connected components $c_{T,i}$ of c_T by using one step BFS.

      Find set $\mathbf{x}_T$ similar with Eq. 2.1 by using one step BFS.

      $r = \max_{c_v \in V} d_V(c_v, c_T)$

      if $r \leq 10e^{-2}$ **then**

        STOP

      Find the farthest point from V_r

      Add $c_{T+1} = \arg\max_{c_v \in V} dr(c_v, V_r)$ to V_r

This modeling procedure is known as sparse coding. Our aim is to reduce the original surfaces dimension with the over-complete dictionary and find a linear combination of the dictionary bases to reconstruct the original surface statistics. The problem statement of sparse coding is described as below.

Given a finite training set of ring-shaped patches (as the description in Sec II. C) $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N) \in \mathbb{R}^{m \times N}$ and $\mathbf{x}_i \in \mathbb{R}^m$, $i = 1, 2, \cdots, N$, where m is the dimension of each ring-shaped patch. In this chapter, we use superscript to represent k-th epoch and use subscript to represent i-th coordinate. We use boldface lower case

letters $\mathbf{x}$ to denote vectors and use boldface upper case letters $\mathbf{X}$ to denote matrices. We then learn dictionary and sparse codes for these input patch features $\mathbf{x}_i$ using sparse coding.

We use $f_i(\cdot)$ to represent the optimization problem of sparse coding for each patch $\mathbf{x}_i$:

$$\min_{\mathbf{D}\in\mathbb{R}^{m\times t},\mathbf{z}_i\in\mathbb{R}^t} f_i(\mathbf{D},\mathbf{z}_i) = \frac{1}{2}||\mathbf{D}\mathbf{z}_i - \mathbf{x}_i||_2^2 + \lambda||\mathbf{z}_i||_1, \tag{2.4}$$

where λ is the regularization parameter, $||\cdot||_2^2$ is the standard Euclidean norm and $||\mathbf{z}_i||_1 = \sum_{j=1}^{t}|z_{i,j}|$. In Eq. 2.4, each input vector will be represented by a linear combination of a few basis vectors of a dictionary. The first term of Eq. 2.4 is the reconstruction error, which measures how well the new feature represents the input vector. The second term of Eq. 2.4 ensures the sparsity of the learned feature $\mathbf{z}_i$. Each $\mathbf{z}_i$ is often called the sparse code. Since $\mathbf{z}_i$ is sparse, there are only a few entries in $\mathbf{z}_i$ which are non-zero. We call its non-zero entries as its support, i.e., $\text{supp}(\mathbf{z}_i) = z_{i,j} : z_{i,j} \neq 0, j = 1,\cdots,t$. $\mathbf{D} = (\mathbf{d}_1,\mathbf{d}_2,\cdots,\mathbf{d}_t)^T \in \mathbb{R}^{m\times t}$ is so called the dictionary, each column represents a basis vector.

Specifically, suppose there are t atoms $\mathbf{d}_j \in \mathbb{R}^m, j = 1,2,\cdots,t$, where the number of atoms is much smaller than n (the number of image patches) but larger than m (the dimension of the image patches). $\mathbf{x}_i$ can be represented by $\mathbf{x}_i = \sum_{j=1}^{t} z_{i,j}\mathbf{d}_j$. In this way, the m-dimensional vector $\mathbf{x}_i$ is represented by a t-dimensional vector $\mathbf{z}_i = (z_{i,1},\cdots,z_{i,t})^T$ ($\mathbf{Z} = (\mathbf{z}_1,\cdots,\mathbf{z}_N) \in \mathbb{R}^{t\times N}$). To prevent an arbitrary scaling of the sparse codes, the columns $\mathbf{d}_i$ are constrained by $\mathbb{C} \triangleq \{\mathbf{D} \in \mathbb{R}^{m\times t} s.t. \forall j = 1,\cdots,t, \mathbf{d}_j^T\mathbf{d}_j \leq 1\}$. Thus, we use $\mathcal{F}(\cdot)$ to represent the sparse coding problem for $\mathbf{X}$, we then rewrite $\mathcal{F}(\cdot)$ as a matrix factorization problem:

$$\min_{\mathbf{D}\in\mathbb{C},\mathbf{Z}} \mathcal{F}(\mathbf{D},\mathbf{Z}) \equiv \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{D},\mathbf{z}_i) = \frac{1}{2}||\mathbf{X} - \mathbf{D}\mathbf{Z}||_F^2 + \lambda||\mathbf{Z}||_1. \tag{2.5}$$

where $||\cdot||_F$ is the Frobenius norm. Eq. 2.5 is a non-convex problem. However, it is a convex problem when either $\mathbf{D}$ or $\mathbf{Z}$ is fixed. When the dictionary $\mathbf{D}$ is fixed, solving

each sparse code $\mathbf{z}_i$ is a Lasso problem (Tibshirani, 1994). Otherwise, when the $\mathbf{Z}$ are fixed, it will become a simple quadratic problem. ODL (Mairal *et al.*, 2009) is known as a state-of-the-art algorithm to solve the sparse coding problem. However, it is relative time consuming due to 1) spend too much time on training a single sample 2) lose the information of sparse codes in the previous epoch. Therefore, it is necessary to find a way to efficiently learn dictionary and save running time. Here we propose our hyperbolic stochastic coding algorithm. It overcomes the above two drawbacks and has two advantages: 1) when updating sparse codes, it only takes a few steps of Coordinate Descent (CD) to generate new sparse codes based on the features of last epoch; 2) when updating dictionaries, it only updates the support vectors (non-zero element in sparse codes). Due to these changes, HSC can dramatically reduce the computational cost of the sparse coding while keeping a comparable performance.

It is known that solving the sparse coding problem is usually very time consuming especially when dealing with large-scale data sets and large size dictionaries (Lee *et al.*, 2006). The proposed algorithm aims to dramatically reduce the computational cost of the sparse coding while keeping the comparable performance (Fig. 2.1 (f)).

We detail our algorithm in the following. Initialize the dictionary via FBS algorithm and denote it as D_1^1. Initialize the sparse code $\mathbf{z}_i^0 = 0$ for $i = 1, \cdots, n$. Here we use superscript k to represent the number of epochs (a cycle of iteratively updating $\mathbf{Z}$ and $\mathbf{D}$) and subscript i to represent the index of data points. Then starting from $k = 1$ and $i = 1$, we do the following:

1. Get an input vector $\mathbf{x}_i$

2. Update z_i^k via one or a few steps of CD (Wu and Lange, 2008):

$$\mathbf{z}_i^k = \mathrm{CD}(\mathbf{D}_i^k, \mathbf{z}_i^{k-1}, \mathbf{x}_i). \tag{2.6}$$

Specifically, for j from 1 to t, we update the j-th coordinate $z_{i,j}^{k-1}$ of $\mathbf{z}_i^{k-1}$ cyclicly as follows:

$$b_j \leftarrow (d_{i,j}^k)^T(\mathbf{x}_i - \mathbf{D}_i^k \mathbf{z}_i^{k-1}) + z_{i,j}^{k-1},$$

$$z_{i,j}^{k-1} \leftarrow h_\lambda(b_j), \mathbf{z}_i^k \leftarrow \mathbf{S}(z_{i,j}^{k-1} - \mathbf{z}_i^{k-1}) + b_j,$$

where $\mathbf{S} = \mathbf{I} - \mathbf{D}^T\mathbf{D}$ and h is the soft thresholding shrinkage function (Combettes and Wajs, 2005a) and λ is the regularization parameter in Eq. 2.5. We call 2) as *one step* of CD (Wu and Lange, 2008). The updated sparse code is then denoted by $\mathbf{z}_i^k$. A detailed derivation of CD can be found in Appendix A.

3. Update the dictionary $\mathbf{D}$ by using stochastic gradient descent (SGD) (Bottou, 1998):

$$\mathbf{D}_{i+1}^k = P_{\mathbb{C}}(\mathbf{D}_i^k - \eta_i^k \nabla_{\mathbf{D}_i^k} f_i(\mathbf{D}_i^k, \mathbf{z}_i^k)), \tag{2.7}$$

where P is the shrinkage function, $\mathbb{C}$ is the feasible set of $\mathbf{D}$ and η_i^k is the learning rate of i-th step in k-th epoch. We set the learning rate as an approximation of the inverse of the Hessian matrix $\mathbf{H}$. The gradient of $\mathbf{D}_i^k$ can be obtained by:

$$\nabla_{\mathbf{D}_i^k} f_i(\mathbf{D}_i^k, \mathbf{z}_i^k) = (\mathbf{D}_i^k \mathbf{z}_i^k - \mathbf{x}_i)(\mathbf{z}_i^k)^T.$$

4. $i = i + 1$. If $i > n$, then set $\mathbf{D}_1^{k+1} = \mathbf{D}_{n+1}^k$, $k = k + 1$ and $i = 1$.

We illustrate our algorithmic framework in Fig. 2.4. At each iteration, with a ring-shaped patch $\mathbf{x}_i$, we perform one step of CD to find the supports of the sparse code $\mathbf{z}_i^{k-1}$. Next, we perform a few steps of CD on the supports to obtain a new sparse code $\mathbf{z}_i^k$. Then we update the supports of the dictionary by the second order SGD to obtain a new dictionary $\mathbf{D}_{i+1}^k$.

It is known that updating the sparse code (step 2) is the most time consuming part (Balasubramanian *et al.*, 2013). CD (Wu and Lange, 2008) is known as one of the state-of-the-art method for solving this lasso problem. Given an input vector $\mathbf{x}_i$,

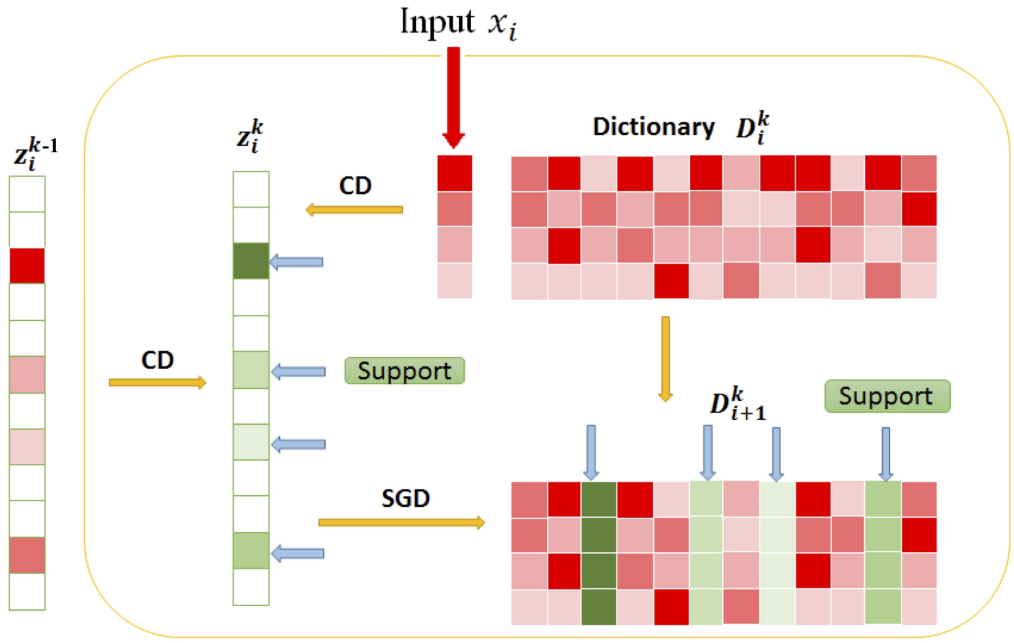


Figure 2.4: Illustration of Hyperbolic Stochastic Coding (HSC) Framework.

CD initializes $\mathbf{z}_i^0 = 0$ and then updates the sparse code many times via matrix-vector multiplication and thresholding. Empirically, the iteration may take tens of hundreds of steps to converge. However, we observe that after a few steps, the supports of the coordinates are very accurate and usually take less then ten steps. Moreover, since the original sparse coding involves an alternating updating, we do not need to run the CD to final convergence during this updating procedure. Therefore, we propose to update the sparse code $\mathbf{z}_i^{k-1}$ by using only a few steps of CD and $\mathbf{z}_i^{k-1}$ is an initial sparse code for updating $\mathbf{z}_i^k$.

After updating the sparse code, we get its supports to update the dictionary. One of our key insights is that we only need to focus on the supports of the dictionary instead of all columns of the dictionary. Let $z_{i,j}^k$ denote j-th entry of $\mathbf{z}_i^k$ and $d_{i,j}^k$ denote the j-th column of the dictionary $\mathbf{D}_i^k$. If $z_{i,j}^k = 0$, then $\nabla_{d_{i,j}^k} f_i(\mathbf{D}_i^k, \mathbf{z}_i^k) = (\mathbf{D}_i^k \mathbf{z}_i^k - \mathbf{x}_i) z_{i,j}^k = 0$. Therefore, $d_{i,j}^k$ does not need to be updated. If $z_{i,j}^k \neq 0$, we can

update $d^k_{i+1,j}$ (the j-th column of the dictionary $\mathbf{D}^k_{i+1}$) as follows:

$$d^k_{i+1,j} \leftarrow d^k_{i,j} - \eta^k_{i,j} \nabla_{d^k_{i,j}} f_i(\mathbf{D}^k_i, \mathbf{z}^k_i) = d^k_{i,j} - \eta^k_{i,j} z^k_{i,j} (\mathbf{D}^k_i \mathbf{z}^k_i - \mathbf{x}_i). \tag{2.8}$$

Note that $\mathbf{z}^k_i$ is a sparse vector, therefore computing $\mathbf{D}^k_i \mathbf{z}^k_i$ is very efficient. $\eta^k_{i,j}$ is the learning rate of the j-column for i-th input in k-th epoch. The computational cost will be significantly reduced when there are limited supports. In contrast, ODL usually has to update all columns of the dictionary. It is because that ODL uses the averaged gradient, which means the supports of the dictionary is itself. Therefore, one has to update all columns of the dictionary and it is time consuming especially when the dictionary size is very large.

When the dataset is very large, the learning rate $\eta^k_{i,j}$ will be very small after going through large number of input vectors. In this case, the dictionary will not change very much and the efficiency of the training will decrease. Therefore, we use an adaptive learning rate in this work. We aim to design a learning rate with the following two principals. The first one is that for different columns of the dictionary, we may use different learning rates. The second is that for the same column, the learning rate should decrease, otherwise the algorithm might not converge. To obtain the learning rate, we use the Hessian matrix of the objective function. It can be shown that the following matrix provides an approximation of the Hessian: $\mathbf{H} = \sum_{k,i} \mathbf{z}^k_i (\mathbf{z}^k_i)^T$, when k and i go to infinity. According to the second order SGD, we should use the inverse matrix of the Hessian as the learning rate.

However, computing a matrix inversion problem is computationally expensive. In order to obtain the learning rate, we simply use the diagonal element of the matrix $\mathbf{H}$. Note that if the columns of the dictionary have low correlation, $\mathbf{H}$ is close to a diagonal matrix. Specifically, we first initialize $\mathbf{H} = 0$. Then update the matrix $\mathbf{H}$ as

follows:

$$\mathbf{H} \leftarrow \mathbf{H} + \mathbf{z}_i^k (\mathbf{z}_i^k)^T, \tag{2.9}$$

when updating the j-th column for the i-th input vector $\mathbf{x}_i$, we replace $\eta_{i,j}^k$ in Eq. 2.8 by $1/h_{jj}$, where h_{jj} is the j-th diagonal element of $\mathbf{H}$. In this way, we do not have to tune the learning rate parameter. It might be worth noting that we do not have to store the whole matrix of $\mathbf{H}$ but only its diagonal elements. We summarize our algorithm in Algorithm 3.

After obtaining features from HSC, max-pooling (Boureau *et al.*, 2010) is adopted on the extracted sparse coding surface features to further reduce feature dimension (Fig. 2.1 (g)). Since we have $n = 2,000$ selected patches (Alg. 2) per subject and each patch with 300 features, it results in 2,000 $\times$ 300 = 600,000 features per subject and $N = 2000 \times$ the number of subjects (Alg. 3). Learning a classifier on small amount subjects with hundred thousands features is prone to over-fitting. Thus, one natural approach is to aggregate statistics of these features at various locations which computes the max value of a particular feature over a region of the surface. These summary statistics are much lower in dimension, and may help reduce over-fitting. Finally, Adaboost (Rojas, 2009) classifier is used for binary classification as shown in Fig. 2.1 (h).

2.4 Convergence Analysis

Here we show that our algorithm is convergent. The objective function $\mathcal{F}$ in Eq. 2.5 can be re-written as follows:

$$\mathcal{F}(\mathbf{D}, \mathbf{z}_1, \cdots, \mathbf{z}_n) = \frac{1}{n} \sum_{i=1}^{n} (\frac{1}{2} \|\mathbf{D}\mathbf{z}_i - \mathbf{x}_i\|_2^2 + \lambda \|\mathbf{z}_i\|_1). \tag{2.10}$$

It is clear that $\mathcal{F}(\mathbf{D}, \mathbf{z}_1, \cdots, \mathbf{z}_n)$ is a nonnegative continuous function over a bounded set $\mathbf{D} \in \mathbb{C}$ and $\|\mathbf{z}_i\| \leq M$ for a real number $M < \infty$, $\mathcal{F}(\mathbf{D}, \mathbf{z}_1, \cdots, \mathbf{z}_n) \rightarrow \infty$ if

Algorithm 3: Hyperbolic Stochastic Coding (HSC)

Input : A collection of overlapped patches

Output: $\mathbf{D} \in \mathbb{R}^{m \times t}$ and $\mathbf{Z} = (\mathbf{z}_1 \cdots \mathbf{z}_n) \in \mathbb{R}^{t \times N}$

begin

  Initialize $\mathbf{D}_1^1$ by selecting random ring-shape patches (Coates and Ng, 2011) from the output of Algorithm 2, $H = 0$, $\mathbf{z}_i^0 = 0$ and $i = 1, \ldots, N$

  for $k = 1$ *to* κ **do**

    for $i = 1$ *to* N **do**

      Get an input vector $\mathbf{x}_i$

      Update $\mathbf{z}_i^k$ via one or a few steps of CD:

        $\mathbf{z}_i^k \leftarrow CD(\mathbf{D}_i^k, \mathbf{z}_i^{k-1}, \mathbf{x}_i)$

      Update the Hessian matrix and the learning rate:

        $\mathbf{H} \leftarrow \mathbf{H} + \mathbf{z}_i^k (\mathbf{z}_i^k)^T, \quad \eta_{i,j}^k = 1/h_{jj}$

      Update the supports of the dictionary via SGD:

        $d_{i+1,j}^k \leftarrow d_{i,j}^k - \eta_{i,j}^k z_{i,j} (\mathbf{D}_i^k \mathbf{z}_i^k - \mathbf{x}_i)$

      if $i = n$ **then**

        $\mathbf{D}_1^{k+1} = \mathbf{D}_{n+1}^k$

$\|\mathbf{z}_i\|_1 \to \infty$. Thus, the minimization problem (Eq. 2.10) has a solution. As the minimization functional $\mathcal{F}$ is not a convex function, the problem (Eq. 2.10) may have multiple solutions and we show our algorithm convergence under certain conditions.

The proof is divided into three parts. We first show the convergence analysis for updating sparse codes (CD step) and updating dictionary (SGD step), respectively. We then combine these two parts to show the convergence of our HSC.

2.4.1 Convergence Analysis of the CD Step

First, we analyze the convergence of CD step. Suppose we update the j-th coordinate $z_{i,j}^k$ of $\mathbf{z}_i^k$ in our cyclic selection approach. It is clear that

$$z_{i,j}^k = \arg\min_{\mathbf{z}} f_i(\mathbf{D}_i^k, z_{i,1}^k, \cdots, z_{i,j-1}^k, z_{i,j}^k, z_{i,j+1}^{k-1}, \cdots, z_{i,t}^{k-1}).$$

Therefore, after going through the whole cycle, we obtain the following result.

Proposition 2.4.1 *For any k and i, we have*

$$f_i(\mathbf{D}_i^k, \mathbf{z}_i^k) \leq f_i(\mathbf{D}_i^k, \mathbf{z}_i^{k-1}). \tag{2.11}$$

Proof. We first use Taylor expansion of $f_i(\mathbf{D}_i^k, \mathbf{z}_i^k)$ at $\mathbf{z}_i^{k-1}$ to rewrite $f_i(\mathbf{D}_i^k, \mathbf{z}_i^k)$ into

$$\begin{aligned} &f_i(\mathbf{D}_i^k, \mathbf{z}_i^{k-1}) + \langle \nabla f_i(\mathbf{D}_i^k, \mathbf{z}_i^{k-1}), \mathbf{z}_i^k - \mathbf{z}_i^{k-1} \rangle + \frac{1}{2}\|\mathbf{D}_i^k(\mathbf{z}_i^k - \mathbf{z}_i^{k-1})\|^2 \\ \leq &f_i(\mathbf{D}_i^k, \mathbf{z}_i^{k-1}) + \langle \nabla f_i(\mathbf{D}_i^k, \mathbf{z}_i^{k-1}), \mathbf{z}_i^k - \mathbf{z}_i^{k-1} \rangle + \frac{\|\mathbf{D}_i^k\|^2}{2}\|\mathbf{z}_i^k - \mathbf{z}_i^{k-1}\|^2 \end{aligned} \tag{2.12}$$

Since $\mathbf{z}_i^k$ is a minimizer of Eq. 2.6, therefore, we have the following equation by Eq. 2.11

$$\begin{aligned} \mathrm{CD}(\mathbf{D}_i^k, \mathbf{z}_i^k, \mathbf{x}_i) = &\langle \nabla f_i(\mathbf{D}_i^k, \mathbf{z}_i^{k-1}), \mathbf{z}_i^k - \mathbf{z}_i^{k-1} \rangle + \frac{L}{2}\|\mathbf{z}_i^k - \mathbf{z}_i^{k-1}\|^2 + \lambda\|\mathbf{z}_i^k\|_1 \\ \leq &\mathrm{CD}(\mathbf{D}_i^k, \mathbf{z}_i^{k-1}, \mathbf{x}_i) = \lambda\|\mathbf{z}_i^{k-1}\|_1, \end{aligned} \tag{2.13}$$

where $L > 0$ is the Lipschitz constant. We then add Eq. 2.12 and Eq. 2.13 together, and simplify the inequality, we have

$$f_i(\mathbf{D}_i^k, \mathbf{z}_i^k) \leq f(\mathbf{D}_i^k, \mathbf{z}_i^{k-1}) - \gamma\|\mathbf{z}_i^k - \mathbf{z}_i^{k-1}\|^2,$$

where $\gamma = (L - \|\mathbf{D}_i^k\|^2)/2 > 0$.

If there are q steps using CD, we should have

$$f_i(\mathbf{D}_i^k, \mathbf{z}_i^{k+q}) \leq f_i(\mathbf{D}_i^k, \mathbf{z}_i^{k-1}). \tag{2.14}$$

□

When q is sufficiently large, we know that $f_i(\mathbf{D}_i^k, \mathbf{z}_i^{k+q})$ decreases to the minimum value $f_i^* = \min_z f_i(\mathbf{D}_i^k, \mathbf{z})$. Since $f_i(\mathbf{D}_i^k, \mathbf{z})$ is a convex function, $f_i^* \leq f_i(\mathbf{D}_i^k, 0) =$

$\frac{1}{2}\|\mathbf{x}_i\|^2 = 1/2$. It follows that $\|\mathbf{z}_i^{k+q}\|_1 \leq \frac{1}{2\lambda}$. This analysis works for all $i = 1, \cdots, n$ and for any k, q. We take enough steps so that $f_i(\mathbf{D}_i^k, \mathbf{z}_i^{k+q}) \leq 1/2$. Therefore, we have the following results for the output $\mathbf{z}_i^k$:

Proposition 2.4.2 *If the number of iterative steps for the CD is sufficiently large, all* $\mathbf{z}_i^k$ *are uniformly bounded for* $i = 1, \cdots, n$ *and* $k \geq 1$.

It might be worth noting that in practice performing only a small number steps of CD is sufficient to guarantee that all $\mathbf{z}_i^k$ are uniformly bounded.

2.4.2 Convergence Analysis of the SGD Step

Second, we study the SGD step. Note that we can always re-index $\mathbf{D}_i^k$ as $\mathbf{D}_{(k-1)n+i+1}$ for convenience. To simplify the notation, we omit the superscript k on the dictionary $\mathbf{D}$ and the learning rate η in this section. Our SGD step in Eq. 2.7 is equivalent to the following two sub-steps by using proximal gradient method (Parikh *et al.*, 2014).

$$\hat{\mathbf{D}}_{i+1} = \arg\min_{\mathbf{D}} g_i(\mathbf{D}_i) + \langle \nabla g_i(\mathbf{D}_i), \mathbf{D} - \mathbf{D}_i \rangle + \frac{1}{2\eta_i}\|\mathbf{D} - \mathbf{D}_i\|_2^2, \tag{2.15}$$

$$\mathbf{D}_{i+1} = P_{\mathbb{C}}(\hat{\mathbf{D}}_{i+1}) = \arg\min_{\mathbf{D}\in\mathbb{C}} \|\hat{\mathbf{D}}_{i+1} - \mathbf{D}\|. \tag{2.16}$$

For simplicity, we let $g_i(\mathbf{D}) \equiv \frac{1}{2}\|\mathbf{D}\mathbf{z}_i^k - \mathbf{x}_i\|_2^2$. Next we show that g_i decreases after performing SGD.

Proposition 2.4.3 $g_i(\hat{\mathbf{D}}_{i+1}) \leq g_i(\mathbf{D}_i)$ *if* $\eta_i \leq \frac{1}{L}$.

Proof. We use Taylor expansion of g_i at $\mathbf{D}_i$ and the Lipschitz differentiation of

$g_i(\mathbf{D})$, we have

$$\begin{aligned} g_i(\hat{\mathbf{D}}_{i+1}) &\leq g_i(\mathbf{D}_i) + \langle \nabla g_i(\mathbf{D}_i), \hat{\mathbf{D}}_{i+1} - \mathbf{D}_i \rangle + \frac{L}{2}\|\hat{\mathbf{D}}_{i+1} - \mathbf{D}_i\|^2 \\ &\leq g_i(\mathbf{D}_i) + \langle \nabla g_i(\mathbf{D}_i), \hat{\mathbf{D}}_{i+1} - \mathbf{D}_i \rangle + \frac{1}{2\eta_i}\|\hat{\mathbf{D}}_{i+1} - \mathbf{D}_i\|^2 \\ &\leq g_i(\mathbf{D}_i) + \langle \nabla g_i(\mathbf{D}_i), \mathbf{D}_i - \mathbf{D}_i \rangle + \frac{1}{2\eta_i}\|\mathbf{D}_i - \mathbf{D}_i\|^2 \\ &= g_i(\mathbf{D}_i), \end{aligned}$$

where the third inequality is due to the optimality condition of Eq. 2.15 and L is Lipschitz constant. □

Proposition 2.4.4 *$g_i(\mathbf{D}_{i+1}) \leq g_i(\mathbf{D}_i)$ if $\eta_i\|\mathbf{z}_i^k\|^2 \leq 1$.*

Proof. Note that $\|\hat{\mathbf{D}}_{i+1} - \mathbf{D}_{i+1}\| \leq \|\hat{\mathbf{D}}_{i+1} - \mathbf{D}_i\|$ by Eq.2.16. By a direct computation of CD in Eq.2.15, we have $\hat{\mathbf{D}}_{i+1} = \mathbf{D}_i - \eta_i \nabla g_i(\mathbf{D}_i)$ because $\hat{\mathbf{D}}_{i+1}$ is the proximation of $\mathbf{D}_i - \eta_i \nabla g_i(\mathbf{D}_i)$. Thus

$$\|\hat{\mathbf{D}}_{i+1} - \mathbf{D}_{i+1}\|^2 = \|\mathbf{D}_i - \eta_i \nabla g_i(\mathbf{D}_i) - \mathbf{D}_{i+1}\|^2 \leq \|\hat{\mathbf{D}}_{i+1} - \mathbf{D}_i\|^2 = \|\eta_i \nabla g_i(\mathbf{D}_i)\|^2.$$

Then, we expand the left side of the above inequality and have

$$\|\mathbf{D}_{i+1} - \mathbf{D}_i\|^2 \leq -2\eta_i \langle \nabla g_i(\mathbf{D}_i), \mathbf{D}_{i+1} - \mathbf{D}_i \rangle. \tag{2.17}$$

According to the definition of g_i, we have

$$\begin{aligned} g_i(\mathbf{D}_{i+1}) &= \frac{1}{2}\|\mathbf{D}_{i+1}\mathbf{z}_i^k - \mathbf{x}_i\|^2 \\ &= \frac{1}{2}\|(\mathbf{D}_{i+1} - \mathbf{D}_i)\mathbf{z}_i^k + \mathbf{D}_i\mathbf{z}_i^k - \mathbf{x}_i\|^2 \\ &= \frac{1}{2}\|\mathbf{D}_{i+1} - \mathbf{D}_i\|^2\|\mathbf{z}_i^k\|^2 + \langle (\mathbf{D}_{i+1} - \mathbf{D}_i)\mathbf{z}_i^k, \mathbf{D}_i\mathbf{z}_i^k - \mathbf{x}_i \rangle + g_i(\mathbf{D}_i) \\ &\leq -\eta_i\|\mathbf{z}_i^k\|^2 \langle \mathbf{D}_{i+1} - \mathbf{D}_i, \nabla g_i(\mathbf{D}_i) \rangle + \langle (\mathbf{D}_{i+1} - \mathbf{D}_i)\mathbf{z}_i^k, \mathbf{D}_i\mathbf{z}_i^k - \mathbf{x}_i \rangle + g_i(\mathbf{D}_i) \\ &= -\eta_i\|\mathbf{z}_i^k\|^2 \langle \mathbf{D}_{i+1} - \mathbf{D}_i, \nabla g_i(\mathbf{D}_i) \rangle + \langle \mathbf{D}_{i+1} - \mathbf{D}_i, \nabla g_i(\mathbf{D}_i) \rangle + g_i(\mathbf{D}_i) \\ &= (1 - \eta_i\|\mathbf{z}_i^k\|^2) \langle \mathbf{D}_{i+1} - \mathbf{D}_i, \nabla g_i(\mathbf{D}_i) \rangle + g_i(\mathbf{D}_i) \end{aligned}$$

By Eq.2.17, we get $\langle \mathbf{D}_{i+1} - \mathbf{D}_i, \nabla g_i(\mathbf{D}_i) \rangle \leq -\frac{1}{2\eta_i}\|\mathbf{D}_{i+1} - \mathbf{D}_i\|^2 \leq 0$. If $\eta_i \leq \frac{1}{\|\mathbf{z}_i^k\|^2}$, we have $g_i(\mathbf{D}_{i+1}) \leq g_i(\mathbf{D}_i)$. □

Since $\|\mathbf{z}_i^k\|_1, i \geq 1, k \geq 1$ are uniformly bounded as in Proposition 2.4.2, $g_i(\mathbf{D}_{i+1}) \leq G$ for a positive constant G independent of i. Furthermore, we have $\nabla g_i(\mathbf{D}_i^k) = (\mathbf{D}_i^k \mathbf{z}_i^k - \mathbf{x}_i)(\mathbf{z}_i^k)^\top$ and it is easy to see that $\|\nabla g_i(\mathbf{D}_i^k)\| \leq C$ for a positive constant C. Thus, we have

Proposition 2.4.5 *Suppose* $\|\nabla g_i(\mathbf{D}_i^k)\|^2 \leq C^2$ *for all* i *and* k, *then* $\|\mathbf{D}_{i+1} - \mathbf{D}_i\| \leq 2\eta_i C$.

Proof. By Eq.2.17, we have

$$\begin{aligned}\|\mathbf{D}_{i+1} - \mathbf{D}_i\|^2 &\leq -2\eta_i \langle \nabla g_i(\mathbf{D}_i), \mathbf{D}_{i+1} - \mathbf{D}_i \rangle \\ &\leq 2\eta_i \|\nabla g_i(\mathbf{D}_i)\| \, \|\mathbf{D}_{i+1} - \mathbf{D}_i\| \\ &\leq 2\eta_i C \|\mathbf{D}_{i+1} - \mathbf{D}_i\|.\end{aligned}$$

The conclusion follows dividing both sides by $\|\mathbf{D}_{i+1} - \mathbf{D}_i\|$ if it is not zero. If it is zero, then the inequality in the proposition is obviously valid. □

One can further prove that $\mathbf{D}_i$ is square summable, we then have the following

Corollary 2.4.1

$$\sum_{i=1}^{\infty} \|\mathbf{D}_i - \mathbf{D}_{i+1}\|^2 \leq 4C^2 \sum_{i=1}^{\infty} \frac{1}{(i+1)^2} < 4C^2.$$

Proof. Indeed, we have

$$\begin{aligned}\|\mathbf{D}_{i+1} - \mathbf{D}_i\|^2 &\leq 2\eta_i \|\nabla g_i(\mathbf{D}_i)\| \, \|\mathbf{D}_{i+1} - \mathbf{D}_i\| \\ &\leq 2C^2 \eta_i^2 + \frac{1}{2} \|\mathbf{D}_{i+1} - \mathbf{D}_i\|^2\end{aligned}$$

by using Cauchy-Schwarz inequality. $\|\mathbf{D}_{i+1} - \mathbf{D}_i\|^2 \leq 4C^2 \eta_i^2$. Summing over $i \geq 1$ concludes the desired inequality. □

2.4.3 Convergence Analysis of the HSC

Now we can give the critical decreasing proposition.

Proposition 2.4.6 *For any epoch $k \geq 1$, we have*

$$\frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{D}_{i+1}^k, \mathbf{z}_i^k) \leq \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{D}_i^k, \mathbf{z}_i^{k-1}).$$

Proof. By Proposition 2.4.1 and Proposition 2.4.4, we have

$$\begin{aligned} f_i(\mathbf{D}_i^k, \mathbf{z}_i^k) &\leq f_i(\mathbf{D}_i^k, \mathbf{z}_i^{k-1}), \\ f_i(\mathbf{D}_{i+1}^k, \mathbf{z}_i^k) &\leq f_i(\mathbf{D}_i^k, \mathbf{z}_i^k). \end{aligned}$$

Combining these two inequities and summing from $i = 1$ to n, we get the desired inequality. □

Proposition 2.4.7 *Let $a_k, k \geq 1$ be a positive sequence. If $a_k, k \geq 1$ satisfy the following*

$$a_{k+1} \leq a_k + \frac{1}{k^{1+\epsilon}}, \quad \forall k \geq 1,$$

then the sequence $a_k, k \geq 1$ converges.

We provide the proof of Proposition 2.4.7 in Appendix B. Now we are ready to give the main result of our HSC convergence analysis.

Theorem 2.4.1 *Suppose $\mathbf{D}^* \in \mathbb{C}$ is a local minimizer such that the mean value $E(\nabla g_i(\mathbf{D}^*)) = 0$ over random variables $(\mathbf{z}_i, \mathbf{x}_i)$ according to Corollary 2.4.1 and Proposition 2.4.4 such that*

$$\begin{aligned} M_k = \frac{1}{n}\sum_{i=1}^{n}(&\frac{1}{2}\|(\mathbf{D}_i^k - \mathbf{D}^*)\mathbf{z}_i^{k-1}\|^2 - \frac{1}{2}\|(\mathbf{D}_{i+1}^k - \mathbf{D}^*)\mathbf{z}_i^k\|^2 \\ &+ \langle\langle \nabla g_i(\mathbf{D}^*), \mathbf{D}_i^k - \mathbf{D}^*\rangle - \langle \nabla g_{i+1}(\mathbf{D}^*), \mathbf{D}_{i+1}^k - \mathbf{D}^*\rangle) = O(1/k^{1+\epsilon}). \end{aligned}$$

as $k \to \infty$ for $\epsilon > 0$. $\mathbf{D}_i^k \to \mathbf{D}^*$ in the following fashion $\mathbf{D}_i - \mathbf{D}^* = O(\frac{1}{i})$, as $i \to \infty$, then our algorithm converges.

Proof. We mainly use Proposition 2.4.6. For each epoch k,

$$\frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{D}_{i+1}^k, \mathbf{z}_i^k) \leq \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{D}_i^k, \mathbf{z}_i^{k-1}).$$

When k goes infinity, we can rewrite $\mathbf{D}_i^k = \mathbf{D}_{(k-1)*n+i+1} = \mathbf{D}^* + O(\frac{1}{kn})$ as $k \to \infty$. We have the Taylor expansion of $\sum f_i(\mathbf{D}_{i+1}^k, \mathbf{z}_i^k)$ at D^* as follows:

$$\frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{D}^*, \mathbf{z}_i^k) + \frac{1}{n}\sum_{i=1}^{n}(\frac{1}{2}\|(\mathbf{D}_{i+1}^k - \mathbf{D}^*)\mathbf{z}_i^k\|^2 + \langle \nabla g_{i+1}(\mathbf{D}^*), \mathbf{D}_{i+1}^k - \mathbf{D}^*\rangle).$$

Similar for $\sum f_i(\mathbf{D}_i^k, \mathbf{z}_i^{k-1})$, we have

$$\frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{D}^*, \mathbf{z}_i^{k-1}) + \frac{1}{n}\sum_{i=1}^{n}(\frac{1}{2}\|(\mathbf{D}_i^k - \mathbf{D}^*)\mathbf{z}_i^{k-1}\|^2 + \langle\langle \nabla g_i(\mathbf{D}^*), \mathbf{D}_i^k - \mathbf{D}^*\rangle).$$

Thus, we have

$$\frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{D}^*, \mathbf{z}_i^k) \leq \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{D}^*, \mathbf{z}_i^{k-1}) + M_k,$$

where M_k is the one as in the assumption of this theorem. Given $M_k = O(1/k^{1+\epsilon})$ as $k \to \infty$ for $\epsilon > 0$ and Proposition 2.4.7, the new sequence $\frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{D}^*, \mathbf{z}_i^k)$ converges to limit $\mathcal{F}(\mathbf{D}^*, \mathbf{z}_1^*, \cdots, \mathbf{z}_n^*)$, where $\mathbf{z}_i^*$ is the limit of a sub-sequence of $\mathbf{z}_i^k$ for $i = 1, \cdots, n$ as $\mathbf{z}_i^k, k \geq 1$ are bounded by Proposition 2.4.2.

Therefore, from the above discussion, $\frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{D}_{i+1}^k, \mathbf{z}_i^k) - \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{D}^*, \mathbf{z}_i^k) \to 0$ and when $k \to \infty$, we conclude that $\lim_{k\to\infty} \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{D}_{i+1}^k, \mathbf{z}_i^k) = \mathcal{F}(\mathbf{D}^*, \mathbf{z}_i^*, \cdots, \mathbf{z}_n^*)$ so that our algorithm converge. □

2.5 Experiments

Data for testing the performances of our proposed HSC are obtained from the ADNI database (adni.loni.usc.edu). The ADNI was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI is to test whether biological markers such as serial MRI and positron emission tomography (PET), combined with clinical and neuropsychological assessments can measure the progression of MCI and early AD. Determination of

Table 2.1: Demographic Statistical Information of Dataset I.

Group	Gender (F/M)	Education	Age	MMSE
AD	15/15	15.22±2.61	76.22±7.34	23.07±2.02
MCI	19/26	16.11±2.56	73.86±8.20	26.95±1.34
CU	18/22	17.25±1.90	76.53±6.02	29.11±1.03

sensitive biomarkers aids researchers and clinicians to develop new treatments and monitor their clinical effectiveness, as well as lessen the time and cost of clinical trials. The initial ADNI (ADNI-1) database recruited 800 subjects from over 50 sites across the U.S. and Canada and it has been followed by ADNI-GO and ADNI-2. To date, these three databases have recruited over 1500 adults, ages 55 to 90, consisting of elderly cognitive unimpaired individuals, people with early or late MCI, and people with early AD. The follow up duration of each subject is specified in their corresponding protocols for ADNI-1, ADNI-2 and ADNI-GO. Subjects of ADNI-1 and ADNI-GO had the option to be followed in ADNI-2. For up-to-date information, see www.adni-info.org.

We use two ADNI datasets to validate our system. The same dataset had been used in our prior paper (Shi *et al.*, 2015). Subjects were chosen on the basis of having at least 36 months of longitudinal MRI and FDG-PET data. In dataset I, we study cortical morphometry for tracking AD progression. Dataset I has 115 T1-weighted MRIs from the ADNI-1 (Weiner *et al.*, 2012) baseline dataset, including 30 AD patients, 45 MCI subjects and 40 cognitively unimpaired (CU) subjects (Shi and Wang, 2019). All subjects underwent through Mini-Mental State Examination (MMSE) (Folstein *et al.*, 1975). The demographic statistics with matched gender, education, age and MMSE are shown in Table 2.1.

Table 2.2: Demographic Statistic Information of Dataset II.

Group	Gender (F/M)	Education	Age	MMSE
MCIc	26/45	15.99±2.73	74.77±6.81	26.83±1.60
MCIs	18/44	15.87±2.76	75.42±7.83	27.66±1.57

Studies indicate that ventricular enlargement is an important measure related with AD progression (Shi *et al.*, 2015; Thompson *et al.*, 2004a). In dataset II, we select 133 subjects from the MCI group in the ADNI-1 (Weiner *et al.*, 2012) baseline dataset as (Shi *et al.*, 2015; Zhang *et al.*, 2016a). All subjects have both volumetric MRI and fluorodeoxyglucose positron emission tomography (FDG-PET) data. They including 71 subjects (age: 74.77 ± 6.81) who developed incident AD during the subsequent 36 months, which we call the MCI converter group, and 62 subjects (age: 75.42 ± 7.83 years) who did not during the same period, which we call the MCI stable group. These subjects were chosen on the basis of having at least 36 months of longitudinal data. If a subject developed incident AD more than 36 months after baseline, it was assigned to the MCI stable group. All subjects underwent thorough clinical and cognitive assessment at the time of acquisition, including the MMSE score, Alzheimers disease assessment scale Cognitive (ADAS-COG) (Rosen *et al.*, 1984) and Auditory Verbal Learning Test (AVLT) (Rey, 1964). The demographic statistical information of this dataset is shown in Table 2.2.

2.5.1 ADNI Baseline Cortical Surfaces

Many researches have analyzed that the cortial surface morphometry is a valid imaging biomarker for AD (Shi and Wang, 2019; Thompson *et al.*, 2004b; Chung *et al.*, 2008a). In Dataset I, we apply HSC to analyze cortical morphometry for AD

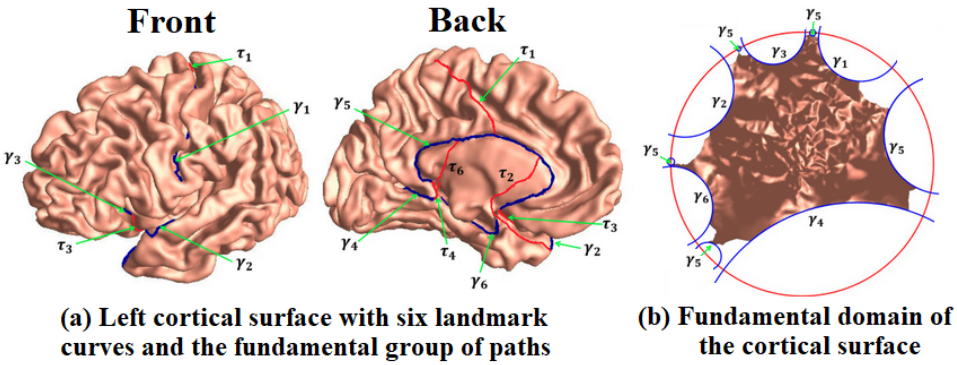


Figure 2.5: Modeling Cortical Surface with Hyperbolic Geometry. (a) Shows Six Identified Open Boundaries, $\gamma_1, \cdots, \gamma_6$. (b) Shows the Hyperbolic Parameter Space, which is the Poincaré Disk Model

related clinical group classification. We use the left hemispheric cerebral cortices and follow (Shi and Wang, 2019) to preprocess cortical surface data. We first use FreeSurfer software (Fischl, 2012) to preprocess the MRIs of 115 subjects and reconstruct their left cortical surfaces. The Caret software (Van Essen, 2012) is then used to automatically label six major brain landmarks, which include the Central Sulcus, Anterior Half of the Superior Temporal Gyrus, Sylvian Fissure, Calcarine Sulcus, Medial Wall Ventral Segment and Medial Wall Dorsal Segment. Fig. 2.5 (a) shows an example of the landmark curves on the left cortical surface, where the six landmark curves are modeled as open boundaries and denoted as $\gamma_1, \cdots, \gamma_6$. The fundamental group of paths are computed by connecting boundary γ_5 to every other boundary and the path is denoted as $\tau_1, \tau_2, \tau_3, \tau_4, \tau_6$. Fig. 2.5 (b) shows that they are embedded into the Poincaré disk. After we cut the cortical surfaces along the delineated landmark curves, the cortical surfaces become genus-0 surfaces with six open boundaries. We finally randomly select the left cortical surface of a CU subject, who is not in the studied subject dataset, as the template surface, and perform the processing steps described in Sec. 2.3.1 and Sec. 2.3.2 to get the hyperbolic surface TBM features.

All experiments are trained for $k = 10$ epochs with a batch size of 1. The regularization parameter λ is set to $0.10 \approx 1.2/\sqrt{m}$, $1/\sqrt{m}$ is a classical normalization

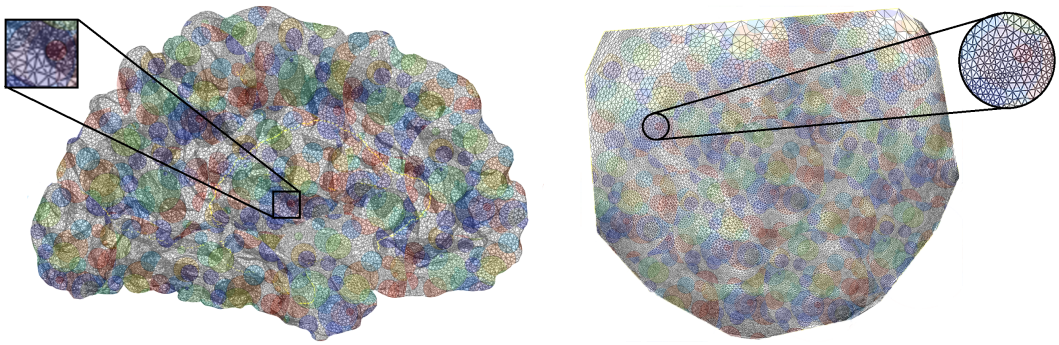


Figure 2.6: Visualization of Computed Image Patches on the Cortical Surface (Left) and Hyperbolic Space (Right). The Zoom-in Pictures Show Some Overlapping Areas between Image Patches.

factor (Bickel *et al.*, 2009) and the constant 1.2 has been shown to produce about 10 non-zero coefficients. We select $n = 2,000$ ring-shaped patches as shown in Fig. 2.6 by FBS on the cortical surface and we have $N = 230,000$ ring-shaped patches for dataset I. Fig. 2.6 (right) is the visualization of cortical morphometry on the hyperbolic parameter domain and Fig. 2.6 (left) projects the selected patches on the hyperbolic parameter domain back to the original cortical surface. Our FBS patch selection algorithm can maintain the same topological structure as the parameter domain.

After learning the sparse codes via HSC, we apply max-pooling (Boureau *et al.*, 2010) for further dimension reduction. Finally, we employ the Adaboost (Rojas, 2009) to do the binary classification and distinguish individuals from different groups. Accuracy (ACC), Sensitivity (SEN), Specificity (SPE) and compute Area Under The Curve (AUC) are computed to evaluate classification results. We randomly split the dataset with a ratio of 8:2 and repeat this procedure for 20 times to avoid data bias. We report the average classification results of (1) AD vs. CU, (2) AD vs. MCI, (3) MCI vs. CU and (4) whole dataset I shown in Table 2.3. For whole dataset I (multi-class classification), we compute AD vs. others, MCI vs. others and CU vs. others, the multi-class classification result is the average of above three group results.

Table 2.3: Classification Results on Dataset I.

Group	ACC	SEN	SPE	AUC
AD vs. CU	1.000	1.000	1.000	1.000
AD vs. MCI	1.000	1.000	1.000	1.000
MCI vs. CU	0.941	0.917	0.900	0.914
Whole Dataset I	0.974	0.938	1.000	0.980

In our prior work (Shi and Wang, 2019), we have shown that the hyperbolic surface features are significantly associated with the diagnostic disease severity. However, it is difficult to directly use hyperbolic surface features for different stages of disease diagnosis classification due to the large amount of features and limited subject numbers. Table 2.3 shows that HSC overcomes the above issue and FBS has a good generalization capability to capture the meaningful features from ring-shaped patches. HSC works well on even more subtle difference classification problem (CU vs. MCI) compared with AD vs. CU. The results on multi-class classification (Whole Dataset I) with 97.4% accuracy, 93.8% sensitivity and 100.0% specificity, show that our new framework make meaningful and high performances on different groups and may be useful for AD diagnosis and prognosis researches.

2.5.2 MCI Converter vs. MCI stable Subjects

In Dataset II, we try to use ventricular morphometry features to discriminate between MCIc and MCIs subjects. To extract hyperbolic surface features, we automatically segment lateral ventricular volumes with the multi-atlas fluid image alignment (MAFIA) method (Chou *et al.*, 2010) from each MRI scan. We then use a topology-preserving level set method (Han *et al.*, 2009) to build surface models and the marching cube algorithm (Lorensen and Cline, 1987) is applied to construct tri-

Table 2.4: Computational Time (hours) and Objective Function (OF) Values of the ODL (Mairal *et al.*, 2009) and HSC for Different Dictionary Sizes.

Methods	Steps	1000	1500	3000
ODL	Updating Z	5.40	15.28	38.75
	Updating D	11.07	40.34	73.59
	Total	16.47	55.62	112.34
	OF Value	0.298	0.270	0.244
HSC	Updating Z	0.158	0.235	0.550
	Updating D	0.030	0.033	0.043
	Total	0.188	0.278	0.593
	OF Value	0.299	0.270	0.245

angular surface meshes (Fig. 3.1 (b)). After the topology optimization, we apply hyperbolic Ricci flow method and conformally map the ventricular surface to the Poincaré disk (Shi *et al.*, 2015). We finally compute the surface TBM features (Shi *et al.*, 2015) and smooth them with surface heat kernel method (Chung *et al.*, 2005).

For HSC, we use the same experimental settings as Sec. 2.5.1. We select $n = 2,000$ ring-shaped patches (Fig. 2.3) by FBS on each side of ventricle for each subject and finally have $N = 532,000$ ring-shaped patches. We have implemented the proposed FBS in matlab 2016a and HSC in C++, all the experiments have been run on a single-GPU, four-core 3.10 Ghz computer. We evaluate the computational efficiency and the classification accuracy on dataset II.

Computational Efficiency

Comparisons of computational time as well as objective function values are given in Table 2.4. We show the time to update the dictionary D and the sparse code Z, respectively, together with the total running time. Table 2.4 reports the computational time on three different dictionary sizes, i.e., 1000×300, 1500×300 and 3000×300. Note that when the size of the dictionary increases, the computational time of ODL (Mairal *et al.*, 2009) increases rapidly. However, for HSC the computational time increases much slower compared to ODL, especially on updating the dictionary. The speedup of updating Z is from 34 times up to 70 times when we increase the dictionary size. Therefore, HSC has a better scalability when dealing with large size dictionaries. In addition, it is worth noting that HSC archives comparable objective function values with ODL. When the dictionary sizes increase, the objective function values decrease, indicating that the dictionary representation ability improves. The convergence analysis and the computational efficiency analysis demonstrate HSC is one potential strategy to apply hyperbolic TBM statistics in the classification analysis of AD diagnosis and prognosis.

Classification Results

We follow the same classification settings as Sec. 2.5.1. We report the average classification accuracy based on 20-times results. Besides, we also compare our work with some other measures and methods. We compute bilateral ventricular volumes and surface areas, which are used as MRI biomarkers in AD research. We also compare HSC with a ventricular surface shape method in (Ferrarini *et al.*, 2008b) (*Shape*), which builds automatically generate comparable meshes of all ventricles. The deformations based morphometry model are employed with repeated permutation tests

Table 2.5: Classification Results on Dataset II.

Method	Region	ACC	SEN	SPE	AUC
HSC	Left	0.727	0.786	0.684	0.754
	Right	0.608	0.625	0.571	0.567
	Whole	**0.967**	**0.933**	**1.000**	**0.976**
Shape (Ferrarini *et al.*, 2008b)	Left	0.535	0.615	0.412	0.572
	Right	0.512	0.515	0.500	0.526
	Whole	0.605	0.656	0.500	0.656
Volume	Left	0.558	0.571	0.552	0.532
	Right	0.517	0.536	0.467	0.430
	Whole	0.535	0.607	0.400	0.452
Area	Left	0.558	0.552	0.571	0.626
	Right	0.465	0.625	0.370	0.493
	Whole	0.512	0.482	0.563	0.517

and then used as geometry features. With our ventricle surface registration results, we follow the *Shape* work (Ferrarini *et al.*, 2008b) for selecting biomarkers and use support vector machine for classification on the same dataset. We test *HSC, Shape, volume and area* measures on the left, right and whole ventricle, respectively. Table 2.5 shows classification performances of four methods. From the experimental results, we can find that the best accuracy (96.7%), the best sensitivity (93.3%) and the best specificity (100%) are achieved when we use TBM features on ventricle hyperbolic space of both sides (whole) for training and testing. The comparison shows that our new framework selects better features, and achieves better and more meaningful classification results.

2.6 Summary

This work presents our initial efforts to develop efficient machine learning methods to work with brain sMRI features computed from general topological surfaces. We validate our proposed HSC and FBS methods on two datasets and the preliminary experimental results demonstrate that the proposed algorithms outperform some other works on both computational running time and classification accuracy. By reducing the dimension of hyperbolic TBM features with the novel HSC algorithm, the present study is capable of applying the low-dimensional HSC measures to diagnose AD and its prodromal stages. In dataset I, the proposed system has an outstanding performance to discriminate the cortical HSC measures of AD, MCI and CU groups (accuracy$>$ 94%). In dataset II, the proposed system outperforms ODL on computational efficiency. It is more than 50 times faster than ODL, and successfully distinguishes the ventricular HSC measures of MCIc subjects from MCIs subjects with a higher accuracy ($> 96\%$) than the classification systems using ventricular volume, area and surface-based biomarkers. These experimental results are consistent with our hypothesis that the lower-dimensional TBM statistics (or HSC measures) may outperform volume, area and shape-based structural measures on discriminating kinds of symptomatic groups related with AD. We also applied our proposed method on early MCI and late MCI in our recent work Zhang *et al.* (2017b). Our method achieved 84% accuracy on ADNI2-dataset with 37 LMCI and 73 EMCI.

There are two important caveats when applying the proposed framework to AD diagnosis and prognosis. First, because of the overlapping patch selection and Max-Pooling scheme, we generally cannot visualize the selected features and it decreases the comprehensibility although we may always visualize statistically significant regions in our prior group difference studies (Shi *et al.*, 2013; Wang *et al.*, 2013). How-

ever, our recent work (Zhang *et al.*, 2018a) made some progress which may potentially better address this problem. Instead of randomly selecting patches to build the initial dictionary, we used group lasso screening to select the most significant features first. Therefore the features used in sparse coding may be visualized on the surface map. In the future, we will incorporate this idea into the proposed framework to improve its interpretation ability. Second, our current work, similar to several other work (e.g. Fan *et al.*, 2007; Colliot *et al.*, 2008; Klöppel *et al.*, 2008; Gerardin *et al.*, 2009; Magnin *et al.*, 2009; Cuingnet *et al.*, 2011; Liu *et al.*, 2011; Shen *et al.*, 2012; Ben Ahmed *et al.*, 2015), uses clinical diagnoses as the "ground truth" diagnoses for training and cross-validation. However, some recent work (e.g. Beach *et al.*, 2012), has reported that neuropathological diagnoses only have limited accuracy values (e.g. only 80 - 90% of the labels are correct) when confirmed with AD histopathology. Under this limitation, we should be cautious when making inference and conclusions on our work for the AD diagnosis since our discovered features are not necessarily real AD biomarkers. Even so, our recent work (Wu *et al.*, 2018) has studied hippocampal morphometry on a cohort consisting of $A\beta$ positive AD ($N = 151$) and matched $A\beta$ negative cognitively unimpaired subjects ($N = 271$) where $A\beta$ positivity was determined using mean-cortical standard uptake value ratio (SUVR) with cerebellum as the reference region over the amyloid PET images. With our Euclidean SCC work (Zhang *et al.*, 2016b) integrating the proposed HSC and MP methods, we achieved an accuracy rate of 90.48% in this task (Wu *et al.*, 2018). The results demonstrate that our proposed framework may potentially help discover pathology-confirmed AD biomarkers.

Chapter 3

MULTI-RESEMBLANCE MULTI-TARGET LOW-RANK CODING

3.1 Introduction

In preclinical AD research, cognitive concerns correlate with structural magnetic resonance imaging (sMRI)-based measures (Frisoni *et al.*, 2010b) of atrophy in several structural measures, including whole-brain, entorhinal cortex, hippocampus and temporal lobe volumes. These findings support their potential usage as predictors of disease progression. However, a notoriously challenging problem in neuroimaging arises from the fact that the imaging feature dimensionality is intrinsically high while only a small number of samples are available. Recent work shows that sparse coding (SC) (Mairal *et al.*, 2009; Li *et al.*, 2017; Zhang *et al.*, 2017c, 2016a) allows us to represent the primary image features as a small set of sparse coefficients and boosts their prediction power. However, the optimization of such problems is extremely time-consuming and the local features with similar descriptors lead to inconsistent sparse codes which may downgrade the statistical power on AD prediction. In addition, modeling sequential longitudinal data by SC is even more challenging because it is hard to find a correlation pattern among images from different time points.

Many multi-task researches are aim to excavate the correlations among data from different modalities or time points. Wang *et al.* (Wang *et al.*, 2011a) propose a multi-task sparse regression and feature selection method to jointly analyze the clinical and neuroimaging data in prediction of the memory performance (Brand *et al.*, 2019). Zhang *et al.* (2012) exploit a $\ell_{2,1}$-norm based group sparse regression method to select features that could be used to jointly predict two clinical status and represent

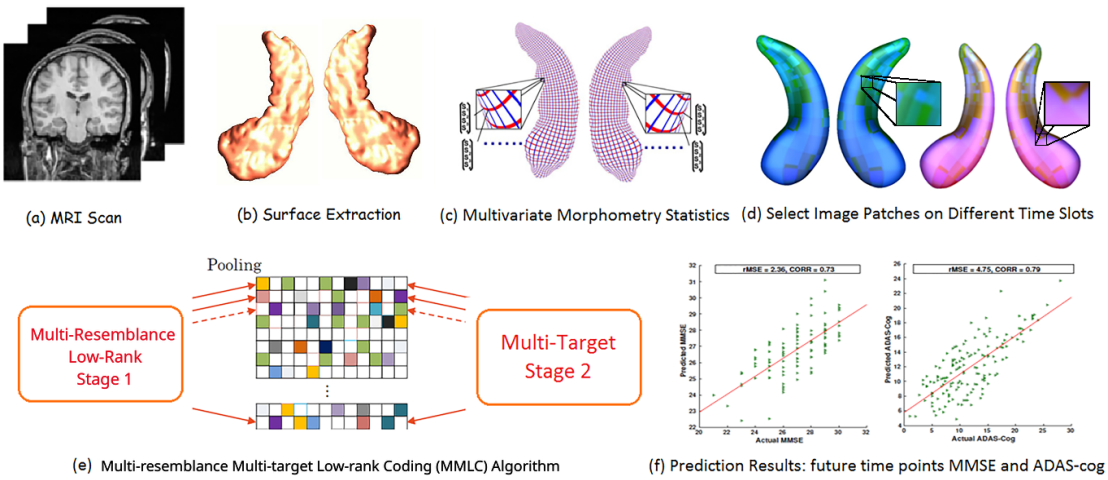


Figure 3.1: The Pipeline of Multi-Resemblance Multi-Target Low-Rank Coding (MMLC) Framework.

the different clinical status. A multi-task sparse learning framework is proposed to integrate multiple incomplete data sources in (Yuan *et al.*, 2012), e.g. there are lots of missing sMRI images in some time points. Our prior work (Zhang *et al.*, 2017c) is a novel unsupervised multi-task SC method which learns the different tasks simultaneously and utilizes shared and task-specific dictionaries to encode both consistent and individual imaging features for longitudinal image data analysis.

Although the multi-task SC may model sequential longitudinal data, the conventional SC method remains computational challenges. We therefore consider the low-rankness in the sparse codes computation that favors both feature sparsity and learning efficiency. Meanwhile, our prior work (Zhang *et al.*, 2017c) simply concatenates the longitudinal data while neglecting the intrinsic resemblance of the longitudinal data. It ignores the fact that the neighborhood features not only have resemblant codebooks but also have resemblant representations. Therefore there is a huge sacrifice of valuable neighborhood time points information from the longitudinal data. To remedy this problem, here we exploit the resemblance among features lying in the neighboring time points and seek an accurate joint representation of these local

features. We design a resemblance penalty term which may make the coefficients of multiple neighboring time points resemblant, ensuring higher correlations between features of near time points than those of distant time points.

The unsupervised multi-task learning overcomes the incomplete source data problem to obtain the sparse features, but the missing clinical label problem is also ubiquitous. It results in multi-task target values after sparse features are extracted. A forthright method is to perform linear regression at each task and determine weighted matrix separately. However, such method treats all tasks independently and ignores the useful information reserved in the change among different tasks and cause strong bias to predict multiple target outputs. Another simple strategy is to remove all patients with missing target values. It, however, significantly reduces the number of samples. Zhou *et al.* (2012) consider multi-task with missing target values in the training process, but the algorithm did not incorporate multiple sources data. For a complete solution, we therefore consider both multiple task incomplete data and multiple outputs with missing target values in this work for exploring the disease prediction problem.

In this chapter, we propose a novel two-stage framework, termed Multi-Resemblance Multi-Target Low-rank Coding (MMLC) algorithm. In stage one, we utilize shared and task-specific dictionaries to encode both consistent and changing imaging features along longitudinal time points and mine the correlations among a small amount features to obtain more consistent sparse codes than learning each time point individually. Meanwhile, we encourage using only a few sparse codebook representations to represent neighboring resemblant features to improve the smoothness of prediction over the longitudinal neighboring time points and maintain a low computational cost. In stage two, we deal with missing clinical label on the target side, thus, we consider both input and target sides' incomplete data in the longitudinal

learning process. MMLC is computed by solving a online low-rank dictionary learning optimization problem, which comprises a sequence of closed-form update steps. They are achieved by the Inexact Augmented Lagrange Multiplier (IALM) that guarantees a fast convergence. Our extensive experimental results on the Alzheimer's Disease Neuroimaging Initiative (ADNI) I cohort (Weiner *et al.*, 2012) show the proposed MMLC achieves significant faster running speed and lower estimation errors, as well as reasonable smooth prediction scores when comparing with six other algorithms, which demonstrates great potential benefits for medical imaging research community.

Our prior work (Zhang *et al.*, 2017c) established the multi-source multi-target dictionary learning framework. The current extended journal manuscript has four major expansions over its conference version, including 1) adding low-rank technique to reduce the dictionary learning computational cost, 2) considering sparse codes of neighboring time point longitudinal features to be resemblant to each other, 3) providing a detailed sequence of closed-form updating steps and theoretical guarantee of a fast convergence, and 4) expanding the experiments to provide additional insights into the benefit of our new method.

3.2 Methods

The pipeline of MMLC is illustrated in Fig. 3.1. We will detail each step in this section. The pipeline source code is publicly available at `http://gsl.lab.asu.edu/software/MMLC`.

3.2.1 Problem Definition and Preliminaries

Given subjects from T time points: $\{\mathbf{X}^1, \cdots, \mathbf{X}^T\}$, our goal is to learn a set of sparse codes $\{\mathbf{S}^1, \cdots, \mathbf{S}^T\}$ for each time point. The sparse code $\mathbf{S}^t \in \mathbb{R}^{m^t \times n^t}$ is a sparse representation of the original input $\mathbf{X}^t \in \mathbb{R}^{p \times n^t}$ and $t \in \{1, ..., T\}$, where p

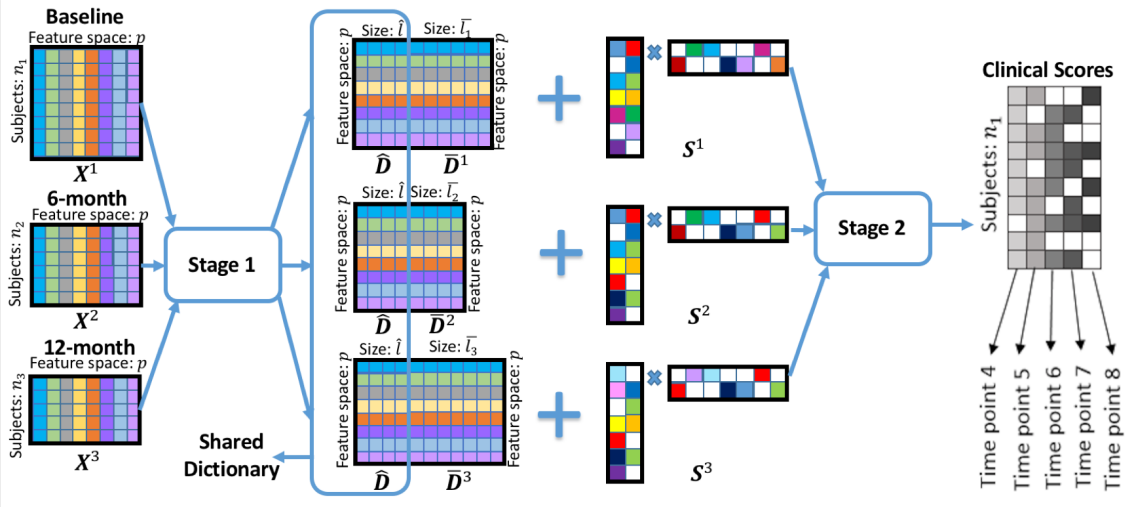


Figure 3.2: Illustration of the Learning Process of MMLC on ADNI-I Cohort from Multiple Different Time Points to Predict Multiple Future Time Points Clinical Scores. In the Figure, There are Three Input Feature Spaces from Baseline, 6-month and 12-month as $\{\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3\}$. We Learn the Dictionaries and Sparse Codes in Stage 1. The Dictionaries have Two Components (Shared Dictionary $\hat{\mathbf{D}}$ and Task-Specific Dictionary $\bar{\mathbf{D}}^t$ Corresponding to Specific Input $\mathbf{X}^t$). The Sparse Codes are Low-rankness and Have Different Resemblance between Each Others (e.g., $\mathbf{S}^1$, $\mathbf{S}^2$ and $\mathbf{S}^2$, $\mathbf{S}^3$ Share Higher Resemblance, i.e., More Common Colors, than $\mathbf{S}^1$, $\mathbf{S}^3$). In Stage 2, We Use Multi-Target Learning to Predict Multiple Target Clinical Scores while Dealing with Missing Label Problem.

is the feature dimension of each sample of $\mathbf{x}_i^t$, $i = 1, ..., n^t$ and n^t is the number of samples for $\mathbf{X}^t$ and m^t is the dimension of each sparse code in $\mathbf{S}^t$.

When employing the conventional single-task sparse coding (SC) to learn the sparse codes $\mathbf{S}^t$ by $\mathbf{X}^t$ individually, we obtain a set of dictionary $\{\mathbf{D}^1, \cdots, \mathbf{D}^T\}$ without correlation between each learnt dictionary. The objective function of single-task SC for time point t will be

$$\min_{\mathbf{D}^t, \mathbf{S}^t} \frac{1}{2}||\mathbf{X}^t - \mathbf{D}^t\mathbf{S}^t||_F^2 + \lambda_1||\mathbf{S}^t||_{1,1}, \ s.t. \mathbf{D}^t \in \Psi^t, \tag{3.1}$$

where $\Psi^t = \{\mathbf{D}^t \in \mathbb{R}^{m^t \times p} : \forall j \in 1, ..., p, ||\mathbf{D}_j^t||_2 \leq 1\}$ and λ_1 is an non-negative parameter. Ψ^t is to prevent an arbitrary scaling of the sparse code, each column of $\mathbf{D}^t$ is restricted to be in a unit ball, i.e., $||\mathbf{D}_j^t|| \leq 1$. The details of SC can be summarized into Algorithm 4.

3.2.2 Multi-Resemblance Low-Rank Sparse Coding Stage

However, single-task SC (Eq. (3.1)) only uses one dictionary $\mathbf{D}$ which is not sufficient to model the variations among subjects from different time points. To address this problem, we integrate the idea of multi-task learning (Liu *et al.*, 2009a) into the SC method. Different from previous works, we propose to learn the intrinsic low-dimensional space of the original data by simultaneously conducting the dictionary learning and sparse feature learning processes. The objective function of our proposed multi-task low-rank SC framework is as follows:

$$\min_{\mathbf{D}^t \in \Psi^t, \mathbf{S}^t} \sum_{t=1}^{T} (\frac{1}{2}||\mathbf{X}^t - \mathbf{D}^t\mathbf{S}^t||_F^2 + \lambda_1||\mathbf{S}^t||_{1,1}), \ s.t.\ rank(\mathbf{S}^t) \leq l^t, \tag{3.2}$$

where the rank l^t-estimate of $\mathbf{S}^t$ denotes as $rank(\mathbf{S}^t) \leq l^t$.

However, Eq. (3.2) dose not consider the correlation between the samples among the multiple time points. Therefore, we proposed to use common and task-specific

Algorithm 4: Single-Task Sparse Coding (STSC)

Input : $\mathbf{X^t}, t = 1, \cdots, T$.

Output: $\mathbf{D}^t$ and $\mathbf{S}^t$, $t = 1, \cdots, T$.

```
begin
    for k = 1 → κ do
        for t = 1 → T do
            Get an input matrix X^t;
            Update S^t by cyclic coordinate descent (CCD) (Canutescu and
             Dunbrack, 2003);
            Update D^t by stochastic gradient descent (SGD) (Zhang, 2004);
            Normalize each column of dictionary D^t .
```

dictionary structure to learn dictionary atoms across multiple time points to capture the correlations. For each input matrix $\mathbf{X}^t$, we learn the dictionary atoms $\mathbf{D}^t$ which are composed of two parts: $\mathbf{D}^t = [\hat{\mathbf{D}}^t, \bar{\mathbf{D}}^t]$ where $\hat{\mathbf{D}}^t \in \mathbb{R}^{\hat{m}\times p}$, $\bar{\mathbf{D}}^t \in \mathbb{R}^{\bar{m}^t\times p}$ and $\hat{m} + \bar{m}^t = m^t$. $\hat{\mathbf{D}}$ is the common dictionary atoms among different tasks and $\hat{\mathbf{D}} = \hat{\mathbf{D}^1} = \cdots = \hat{\mathbf{D}^T}$ while $\bar{\mathbf{D}}^t$ is different from each other and only learned from the corresponding task input matrix $\mathbf{X}^t$. The objective function can be reformulated as follows:

$$\min_{\mathbf{D}^t\in\Psi^t,\mathbf{S}^t} \sum_{t=1}^{T} (\frac{1}{2}||\mathbf{X}^t - [\hat{\mathbf{D}}, \bar{\mathbf{D}^t}]\mathbf{S}^t||_F^2 + \lambda_1||\mathbf{S}^t||_{1,1} + \lambda_2||\mathbf{S}^t||_*). \tag{3.3}$$

where λ_1 and λ_2 quantify the tradeoff between sparsity and low-rankness in the feature learning process. $\lambda_2 = 0$ is the special case of Eq. (3.3), the problem (3.3) will become sparse coding problem. Specifically, the objective function Eq. (3.2) is a non-convex problem due to the non-convexity of the $rank(\mathbf{S})$. We use the convex relaxation technique (Boyd and Vandenberghe, 2004) in Eq. (3.3), the trace norm (nuclear norm) has been known as the convex envelop of the function of the rank $||\mathbf{S}||_* \leq rank(\mathbf{S}), \forall \mathbf{S} \in \mathbb{C} = \{\mathbf{S}|||\mathbf{S}||_2 \leq 1\}$.

The longitudinal data of the time points close to the baseline MR images has higher resemblance than those of time points distant to the baseline MR images (e.g., 3-month and 6-month MR images are more resemblant to baseline images than those of 12-month MR images). We further use a Gaussian similarity kernel to emphasize such inherent resemblance knowledge between two different time points:

$$w_{p,q} = \exp(\frac{-||\mathbf{S}^p - \mathbf{S}^q||}{2\sigma^2}), \tag{3.4}$$

where σ is the standard deviation of the training samples/patches, p and q donate time point p and time point q.

The function $w_{p,q}$ is used to penalize the distance between two time points so that it emphasizes the inherent resemblance, i.e., the nearby time points learn high

resemblance sparse codes $\mathbf{S}$ and distant time points learn high disparities. The final objective function of MMLC stage-I multi-resemblant low-rank SC stage can be formalized as follows:

$$\begin{aligned}\min_{\mathbf{D}^t\in\Psi^t,\mathbf{S}^t} \sum_{t=1}^{T}(\frac{1}{2}||\mathbf{X}^t-[\hat{\mathbf{D}},\bar{\mathbf{D}^t}]\mathbf{S}^t||_F^2+\lambda_1||\mathbf{S}^t||_{1,1}+\lambda_2||\mathbf{S}^t||_*) \\ +\lambda_3\sum_{p=1}^{t-1}\sum_{q=p+1}^{t} w_{p,q}||\mathbf{S}^p-\mathbf{S}^q||_2^2.\end{aligned} \tag{3.5}$$

where λ_3 is a non-negative regularization parameter. We will discuss how to optimize Eq. (3.5) in Sec. 3.3.

Fig. 3.2 illustrates the learning process of MMLC with subjects of ADNI from three different time points which represents as $\mathbf{X}^1$, $\mathbf{X}^2$ and $\mathbf{X}^3$, respectively. Through the multi-resemblant low-rank SC stage (Stage 1), we obtain the dictionary and sparse codes for subjects from each time point t: $\mathbf{D}^t$ and $\mathbf{S}^t$. A dictionary $\mathbf{D}^t$ is composed by a shared dictionary $\hat{\mathbf{D}}^t$ across all tasks and a task-specific part $\bar{\mathbf{D}}^t$ only corresponding with the specific task $\mathbf{X}^t$. As a result, the sparse codes are low-rankness and have different resemblance between each others (e.g., $\mathbf{S}^1$, $\mathbf{S}^2$ and $\mathbf{S}^2$, $\mathbf{S}^3$ share higher resemblance, i.e., more common colors, than $\mathbf{S}^1$, $\mathbf{S}^3$).

3.2.3 Multi-Target Learning with Missing Label Stage

We measure the cognitive scores of patients at multiple time points in the longitudinal AD study. We formulate the prediction of clinical scores at multiple future time points simultaneously rather than considering the prediction of cognitive scores as a set of single time point regression since the intrinsic temporal smoothness information among different tasks can be incorporated into the model as the prior knowledge. However, there are many missing clinical scores at certain time points, especially for 36 and 48 months ADNI data. It is necessary to incorporate the missing target values with multi-task regression to predict clinical scores.

Algorithm 5: Multi-Resemblance Multi-Target Low-rank Coding (MMLC)

Input : Samples $\mathbf{X^t}$ and corresponding labels $\mathbf{Y^t}$ from different time points, epoches κ, $\lambda_1, \lambda_2, \lambda_3$, μ_1, μ_2, γ, ϕ and $\hat{\mathbf{D}} = \mathbf{D}_0$.

Output: The models for different time points $\mathbf{W}^t$.

begin

  Stage I: Multi-Resemblance Low-Rank SC Stage

  for $k = 1 \rightarrow \kappa$ **do**

    for $t = 1 \rightarrow T$ **do**

      For each input matrix $\mathbf{X}^t$;

      Update $\mathbf{S}^{t,(k)}$ via Alg. 6;

      Update $||\mathbf{S}^{t,(k)}||_{1,1}$ and $||\mathbf{S}^{t,(k)}||_*$ by Eq. (3.19) and Eq. (3.20);

      Update $\hat{\mathbf{D}}^{(k)}$: $\hat{\mathbf{D}}^{(k)} = \mathbf{D}_0$ ($\mathbf{D}_0 = \hat{\mathbf{D}}^{(k-1)}$);

      Update the $\hat{\mathbf{D}}^{(k)}$ and $\bar{\mathbf{D}}^{t,(k)}$ via Alg. 7;

      Calculate $w_{p,q}$ function by Eq (3.4);

      Update $\mathbf{S}^{t,(k)}$ by Eq. (3.25);

      $\mathbf{D}_0 = \hat{\mathbf{D}}^{(k)}$;

  Obtain the learnt sparse codes $\mathbf{S}^t$, $t = 1, \cdots, T$.

  Stage II: Multi-Target Regression Stage

  for $t = 1$ *to* T **do**

    Given $\mathbf{Y}_j^t \in \mathbf{Y}^t$, for the jth model $\mathbf{w}_j^t \in \mathbf{W}^t$: $\quad \mathbf{w}_j^t = (\widetilde{\mathbf{S}}^t \widetilde{\mathbf{S}}^{tT} + \xi \mathbf{I})^{-1} \widetilde{\mathbf{S}}^t \widetilde{\mathbf{Y}}_j^t$

In this chapter, we use a matrix $\Theta \in \mathbb{R}^{m^t \times n^t}$ to indicate missing target values, where $\Theta_{i,j} = 0$ if the target value of label $\mathbf{Y}_{i,j}^t$ is missing and $\Theta_{i,j} = 1$ otherwise. Give the sparse codes $\{\mathbf{S}^1, ..., \mathbf{S}^T\}$ and corresponding labels $\{\mathbf{Y}^1, ..., \mathbf{Y}^T\}$ from different times where $\mathbf{Y}^t \in \mathbb{R}^{m^t \times n^t}$, we formulate the multi-target learning stage with missing

target values as:

$$\min_{\mathbf{W}^1,\cdots,\mathbf{W}^T} \sum_{t=1}^{T} ||\Theta(\mathbf{Y}^t - \mathbf{W}^t\mathbf{S}^t)||_F^2 + \xi \sum_{t=1}^{T} ||\mathbf{W}^t||_F^2. \tag{3.6}$$

Although Eq. 3.6 is associated with missing values on the labels, we show that it has a close form and present the theoretical analysis of MMLC stage-II as follows:

Theorem 3.2.1 For the data matrix pair $(\mathbf{S}^t, \mathbf{Y}^t)$, we denote the jth row's labels of $\mathbf{Y}^t$ as $\widetilde{\mathbf{Y}}_j^t$. We represent the remaining data after removing the missing value in $\mathbf{Y}_j^t$ as $\widetilde{\mathbf{S}}^t$ and $\widetilde{\mathbf{Y}}_j^t$. The problem of Eq. (3.6) can be solved sequentially with $\mathbf{w}_j^t = (\widetilde{\mathbf{S}}^t\widetilde{\mathbf{S}}^{tT} + \xi\mathbf{I})^{-1}\widetilde{\mathbf{S}}^t\widetilde{\mathbf{Y}}_j^t$.

Proof Eq. 3.6 can be rewritten as

$$\min_{w_j^t} ||(\widetilde{\mathbf{Y}}_j^t - \mathbf{w}_j^t\widetilde{\mathbf{S}}^t)||_2^2 + \xi||\mathbf{w}_j^t||_2^2 \tag{3.7}$$

It is known as Ridge regression (Hoerl and Kennard, 1970a). To optimize the problem, we calculate the gradient and set the gradient to be zero. Then we can get the optimal $\mathbf{w}_j^t$ by the following steps:

$$\begin{aligned}
&2\widetilde{\mathbf{S}}^t(\widetilde{\mathbf{S}}^{tT}\mathbf{w}_j^t - \widetilde{\mathbf{Y}}_j^t) + 2\xi\mathbf{w}_j^t = 0,\\
&\widetilde{\mathbf{S}}^t\widetilde{\mathbf{S}}^{tT}\mathbf{w}_j^t - \widetilde{\mathbf{S}}^t\widetilde{\mathbf{Y}}_j^t + \xi\mathbf{w}_j^t = 0,\\
&(\widetilde{\mathbf{S}}^t\widetilde{\mathbf{S}}^{tT} + \xi\mathbf{I})\mathbf{w}_j^t = \widetilde{\mathbf{S}}^t\widetilde{\mathbf{Y}}_j^t,\\
&\mathbf{w}_j^t = (\widetilde{\mathbf{S}}^t\widetilde{\mathbf{S}}^{tT} + \xi\mathbf{I})^{-1}\widetilde{\mathbf{S}}^t\widetilde{\mathbf{Y}}_j^t.
\end{aligned}$$

After solving $\mathbf{w}_j^t$ for each time point $j \in \{1, ..., m^t\}$, we obtain the learnt model $\{\mathbf{W}^1, \cdots, \mathbf{W}^T\}$ for prediction. □

3.3 Optimization Analysis

In this section, we explain the update procedures for MMLC. Eq. (3.5) is a non-convex problem. However, it will become a convex problem when we fix either $\mathbf{D}$

or $\mathbf{S}$. When the sparse codes $\mathbf{S}$ is fixed, solving dictionary $\hat{\mathbf{D}}$ and $\bar{\mathbf{D}}$ can be solved as a quadratically constrained quadratic programming (QCQP) problem (Boyd and Vandenberghe, 2004). At the end of each update in MMLC stage-I, we update the shared dictionary Φ: $\Phi = \hat{\mathbf{D}}^t$ and let $\hat{\mathbf{D}}^1 = \cdots = \hat{\mathbf{D}}^t$. When the dictionary $\mathbf{D}$ is fixed, solving each sparse code $\mathbf{s}_i$ can be view as a sparse group Lasso problem (Simon *et al.*, 2013). We alternately update $\mathbf{D}^t$ and $\mathbf{S}^t$ for $k = \kappa$ epoches and summarize the optimization details into Algorithm 5.

In Algorithm 5, for each image patch $\mathbf{x}_i^t$, we learn the i-th sparse code $\mathbf{s}_i^{t,(k+1)}$ from $\mathbf{s}^t$ by several steps of cyclic coordinate descent (CCD) (Canutescu and Dunbrack, 2003). We then use learnt sparse codes $\mathbf{s}_i^{t,(k+1)}$ to update the dictionary $\hat{\mathbf{D}}^{t,(k+1)}$ and $\bar{\mathbf{D}}^{t,(k+1)}$ by one step stochastic gradient descent (SGD) (Zhang, 2004). Since $\mathbf{s}_i^{t,(k+1)}$ is very sparse, we use the index set $\mathbf{I}_i^{t,(k+1)}$ to record the location of non-zero entries in $\mathbf{s}_i^{t,(k+1)}$ to accelerate the update of sparse codes and dictionaries. Φ is updated by the end of the k-th iteration to ensure $\hat{\mathbf{D}}^{t,(k+1)}$ is the same part among all the dictionaries.

3.3.1 Updating the Low-Rankness Sparse Codes

After we pick an image patch $\mathbf{x}_i^t$ from the sample $\mathbf{X}^t$ at the time point t, we fix the dictionary $\mathbf{D}$ and only consider updating the first sparse codes term $\mathbf{S}$. The optimization problem becomes the following form:

$$\min_{\mathbf{S}^t} \sum_{t=1}^{T} (\frac{1}{2}||\mathbf{X}^t - [\hat{\mathbf{D}}, \bar{\mathbf{D}^t}]\mathbf{S}^t||_F^2 + \lambda_1||\mathbf{S}^t||_{1,1}) \tag{3.8}$$

Coordinate descent (Canutescu and Dunbrack, 2003) is known as one of the state-of-the-art methods for solving this Lasso problem (Tibshirani, 1996a). In this study, we perform the CCD to optimize Eq (3.8). Empirically, the iteration may take thousands of steps to converge, which is time-consuming in the optimization process of dictionary learning. However, we observed that after a few steps, the support of the coordinates,

Algorithm 6: Updating sparse codes $\mathbf{s}_i^{t,(k+1)}$

Input : Image patch $\mathbf{x}_i^t$, dictionaries $\hat{\mathbf{D}}^{t,(k)}$ and $\bar{\mathbf{D}}^{t,(k)}$, sparse codes $\mathbf{s}_i^{t,(k)}$ and index set $\mathbb{I}_i^{t,(k)}$.

Output: $\mathbf{s}_i^{t,(k+1)}$ and $\mathbb{I}_i^{t,(k+1)}$.

```
begin
    for j = 1 to p^t do
        Update s_{i,j}^{t,(k+1)} by Eq. (3.9) and Eq. (3.10).
        if s_{i,j}^{t,(k+1)} ≠ 0 then
            Put j into the index set I_i^{t,(k+1)}.
    for j = 1 to Q do
        for l ∈ I_i^{t,(k+1)} do
            Update l by Eq. (3.12) and Eq. (3.13).
```

i.e., the locations of the non-zero entries in $\mathbf{s}_i^t$, becomes very stable, usually after less than ten steps. In this study, we perform P steps CCD to generate the non-zero index set $\mathbb{I}_t^{k+1}$, recording the non-zero entry of $\mathbf{s}_i^{t,(k+1)}$. Then we perform Q steps CCD to update the sparse codes only on the non-zero entries of $\mathbf{s}_i^{t,(k+1)}$, accelerating the learning process significantly. Stochastic coordinate coding (SCC) (Lin *et al.*, 2014) employs a similar strategy to update the sparse codes in a single task. For the multi-task learning, we summarize the updating rules as follows:

(a) Perform P steps CCD to update the locations of the non-zero entries $\mathbb{I}_i^{t,(k+1)}$ and the model $\mathbf{s}_i^{t,(k+1)}$.

(b) Perform Q steps CCD to update the $\mathbf{s}_i^{t,(k+1)}$ in the index of $\mathbb{I}_i^{t,(k+1)}$.

In (a), we will pick up j-th coordinate to update the model $\mathbf{s}_{i,j}^t$ and non-zero entries, where $j \in \{1, ..., p^t\}$ in every CCD step. We perform the update from the 1st

coordinate to the p^t-th coordinate. Meanwhile, we calculate the gradient $\mathbf{g}$ based on Eq. (3.8)) and update the model $\mathbf{s}_{i,j}^{t,(k+1)}$ based on $\mathbf{g}$. The calculation of $\mathbf{g}$ and $\mathbf{s}_{i,j}^{t,(k+1)}$ follows the equations:

$$\mathbf{g} = [\hat{\mathbf{D}}^{t,(k)}, \bar{\mathbf{D}}^{t,(k)}]_j^T(\Omega([\hat{\mathbf{D}}^{t,(k)}, \bar{\mathbf{D}}^{t,(k)}], \mathbf{s}_i^{t,(k)}, \mathbb{I}_i^{t,(k)}) - \mathbf{x}_i^t), \tag{3.9}$$

$$\mathbf{s}_{i,j}^{t,(k+1)} = \Gamma_\lambda(\mathbf{s}_{i,j}^{t,(k)} - \mathbf{g}), \tag{3.10}$$

where Ω is a sparse matrix multiplication function that has three input parameters. Take $\Omega(\mathbf{A}, \mathbf{b}, \mathbb{I})$ as an example, $\mathbf{A}$ is a matrix, $\mathbf{b}$ denotes a vector and $\mathbb{I}$ records the locations of non-zero entries in $\mathbf{b}$ (an index set). The output value of Ω is defined as: $\Omega(\mathbf{A}, \mathbf{b}, \mathbb{I}) = \mathbf{Ab}$. We manipulate the non-zero entries of $\mathbf{b}$ and the corresponding columns of $\mathbf{A}$ based on the index set $\mathbb{I}$ when computing $\mathbf{Ab}$ so that we speed up the calculation by utilizing the sparsity of $\mathbf{b}$. Γ is the soft thresholding shrinkage function (Combettes and Wajs, 2005b) as below:

$$\Gamma_\varphi(x) = sign(x)(|x| - \varphi). \tag{3.11}$$

In the end of (a), we count the non-zero entries in $\mathbf{s}_i^{t,(k+1)}$ and store the non-zero index in $\mathbb{I}_i^{t,(k+1)}$. In (b), we perform Q steps CCD by only considering the non-zero entries in $\mathbf{s}_i^{t,(k+1)}$. As a result, for each index $l \in \mathbb{I}_i^{t,(k+1)}$, we calculate the gradient $\mathbf{g}$ and update the $\mathbf{s}_{i,l}^{t,(k+1)}$ by:

$$\mathbf{g} = [\hat{\mathbf{D}}^{t,(k)}, \bar{\mathbf{D}}^{t,(k)}]_l^T(\Omega([\hat{\mathbf{D}}^{t,(k)}, \bar{\mathbf{D}}^{t,(k)}], \mathbf{s}_i^{t,(k+1)}, \mathbb{I}_i^{t,(k+1)}) - \mathbf{x}_i^t), \tag{3.12}$$

$$\mathbf{s}_{i,l}^{t,(k+1)} = \Gamma_\lambda((\mathbf{s}_{i,l}^{t,(k+1)} - \mathbf{g}). \tag{3.13}$$

Since we only focus on the non-zero entries of the model and P is less than 10 iteration and Q is a much larger number, we significantly accelerate the entire sparse codes learning process. The procedure of updating sparse codes can be summarized into Algorithm 6.

However, in Eq. (3.5), there are two convex and non-smooth regularizers for $\mathbf{S}^t$. We propose to update the low-rankness sparse codes by using the conventional Inexact Augmented Lagrange Multiplier (IALM) (Fernández and Solodov, 2012). IALM is an iterative method that augments the Lagrangian function with quadratic penalty terms, which allows closed-form updates for each variables in the problem. Therefore, solving the ℓ_1 and the nuclear norm will result in solving the following problem, where we use two slack variables $\mathbf{S}_2^t$ and $\mathbf{S}_3^t$ for the two terms:

$$\begin{aligned}\min_{\mathbf{D}^t\in\Psi^t,\mathbf{S}_1^t,\mathbf{S}_2^t,\mathbf{S}_3^t} \sum_{t=1}^{T}(\frac{1}{2}||\mathbf{X}^t-[\hat{\mathbf{D}},\bar{\mathbf{D}^t}]\mathbf{S}_1^t||_F^2+\lambda_1||\mathbf{S}_2^t||_{1,1}+\lambda_2||\mathbf{S}_3^t||_*+\\ tr[L_1(\mathbf{S}_1^t-\mathbf{S}_2^t)]+tr[L_2(\mathbf{S}_1^t-\mathbf{S}_3^t)]+\frac{\mu_1}{2}||\mathbf{S}_1^t-\mathbf{S}_2^t||_F^2+\frac{\mu_2}{2}||\mathbf{S}_1^t-\mathbf{S}_3^t||_F^2),\end{aligned} \tag{3.14}$$

where L_1 and L_2 are lagrange multipliers, and μ_1 and μ_2 are two positive scalars. IALM efficiently minimize Eq. (3.14) and the validity and optimality of Eq. (3.14) is guaranteed by the following theorem.

Theorem 3.3.1 *For Eq.* (3.14)*, if* $\{\mu_r^k\}(r=1,2)$ *is non-decreasing and* $\sum_{k=1}^{+\infty}1/\mu_r^k=+\infty$ *then* $(\mathbf{S}_2,\mathbf{S}_3)$ *converge to an optimal solution* $(\mathbf{S}_2^*,\mathbf{S}_3^*)$.

Proof: The convergence of Eq. (3.14) when $\{\mu_r^k\}$ is upper bounded has been proved by (Lin *et al.*, 2010). Then, Suppose $\mu_r^k\to+\infty$, we have $\sum_{k=1}^{+\infty}\frac{1}{\mu_r^2}{}^k||L^{k+1}-L^k||_F^2<+\infty$. Therefore, $||\mathbf{S}_1-\mathbf{S}_2^k-\mathbf{S}_3^k||_F=1/\mu_r^k||L^k-L^{k-1}||_F\to 0$. Then any accumulation point of $(\mathbf{S}_2,\mathbf{S}_3)$ is a feasible solution.

Let's use $\mathbf{S}^*$ donate the optimal objective value of the Eq. (3.14). As $L_1^k\in\partial(\lambda||\mathbf{S}_2^k||_1)$ and $L_2^k\in\partial||\mathbf{S}_3^k||_*$, $\lambda=\lambda_1/\lambda_2$, $\hat{\mathbf{S}}^*=\mathbf{S}_2^*+\mathbf{S}_3^*$, we have

$$\begin{aligned}&\lambda||\mathbf{S}_2^k||_1+||\mathbf{S}_3^k||_*\le\lambda||\mathbf{S}_2^*||_1+||\mathbf{S}_3^*||_*-<L_1^k,\mathbf{S}_2^*-\mathbf{S}_2^k>-<L_2^k,\mathbf{S}_3^*-\mathbf{S}_3^k>\\ =&\mathbf{S}^*+<L_1^k-L^*,\mathbf{S}_2^k-\mathbf{S}_2^*>+<L_2^k-L^*,\mathbf{S}_3^k-\mathbf{S}_3^*>-<L^*,\mathbf{S}_2^*-\mathbf{S}_2^k+\mathbf{S}_3^*-\mathbf{S}_3^k>\\ =&\mathbf{S}^*+<L_1^k-L^*,\mathbf{S}_2^k-\mathbf{S}_2^*>+<L_2^k-L^*,\mathbf{S}_3^k-\mathbf{S}_3^*>-<L^*,\hat{\mathbf{S}}^*-\mathbf{S}_2^k-\mathbf{S}_3^k>.\end{aligned} \tag{3.15}$$

Due to μ^k is nondecreasing in the assumption, then each entry of the following series is nonnegative and its sum is finite.

$$\sum_{k=1}^{+\infty} 1/\mu_r^k(< L_1^k - L^*, \mathbf{S}_2^k - \mathbf{S}_2^* > + < L_2^k - L^*, \mathbf{S}_3^k - \mathbf{S}_3^* >) < +\infty \tag{3.16}$$

As $1/\mu_r^k \to +\infty$, there must exist a subsequence $(\mathbf{S}_{2_s}, \mathbf{S}_{3_s})$ such that

$$< \mathbf{S}_{2_s}^k - \mathbf{S}_2^*, L_{1_s}^k - L^* > + < \mathbf{S}_{3_s}^k - \mathbf{S}_3^*, \hat{L}_{2_s}^k - L^* > \to 0. \tag{3.17}$$

Then, we have that

$$\lim_{s \to +\infty} \lambda ||\mathbf{S}_{2_s}||_1 + ||\mathbf{S}_{3_j}||_* \leq S^*. \tag{3.18}$$

Therefore, $(\mathbf{S}_{2_s}, \mathbf{S}_{3_s})$ approaches to an optimal solution $(\mathbf{S}_2^*, \mathbf{S}_3^*)$ for the problem Eq. (3.14). □

Theorem 3.3.1 only guarantees convergence but does not specify the rate of convergence for the IALM method and we discuss the convergence rate at the end of this section. we use blockwise coordinate descent to alternatively update each variable of $\mathbf{S}_1^t$, $\mathbf{S}_2^t$, $\mathbf{S}_3^t$ with all other variables fixed to their most recent values as follows:

$$\begin{aligned} &\mathbf{S}_2^{t*} = \Omega_{\frac{\lambda_1}{\mu_1}}(\mathbf{S}_1^t + \frac{L_1}{\mu_1}), \mathbf{S}_3^{t*} = \Theta_{\frac{\lambda_2}{\mu_2}}(\mathbf{S}_1^t + \frac{L_2}{\mu_2}), \\ &\mathbf{S}_1^{t*} = (\mathbf{D}^{tT}\mathbf{D}^t\mu_1\mathbf{I} + \mu_2\mathbf{I})^{-1}\mathbf{G}, \end{aligned} \tag{3.19}$$

where $\mathbf{G} = \mathbf{D}^{tT}\mathbf{X}^t - L_1 - L_2 + \mu_1\mathbf{S}_2^t + \mu_2\mathbf{S}_3^t$, $\Omega_\lambda(\mathbf{S}) = sign(\mathbf{S})(|\mathbf{S}| - \lambda)_+$ is the soft-thresholding operator and $\Theta_\lambda(\mathbf{S}) = U\Omega_\lambda(\mathbf{\Sigma})V^T$ is the singular value soft-thresholding operator with $\mathbf{S} = U\Sigma V^T$ is the SVD of $\mathbf{S}$. Then, we can update the multipliers with $\phi > 1$ as follows,

$$\begin{aligned} &L_1 = L_1 + \mu_1(\mathbf{S}_1^t - \mathbf{S}_2^t); L_2 = L_2 + \mu_2(\mathbf{S}_1^t - \mathbf{S}_3^t); \\ &\mu_1 = \phi\mu_1; \mu_2 = \phi\mu_2. \end{aligned} \tag{3.20}$$

Algorithm 7: Updating Dictionaries $\hat{\mathbf{D}}_t^{k+1}$ and $\bar{\mathbf{D}}_t^{k+1}$

Input : Image patch $\mathbf{x}_i^t$, dictionaries $\hat{\mathbf{D}}^{t,(k)}$ and $\bar{\mathbf{D}}^{t,(k)}$, sparse codes $\mathbf{s}_i^{t,(k+1)}$ and index set $\mathbb{I}_i^{t,(k+1)}$.

Output: The updated dictionaries $\hat{\mathbf{D}}_t^{k+1}$ and $\bar{\mathbf{D}}_t^{k+1}$

```
begin
    Update the Hessian matrix H_t^{k+1} by Eq. (3.22).
    R = Ω([D̂^{t,(k)}, D̄^{t,(k)}], s_i^{t,(k+1)}, I_i^{t,(k+1)}) − x_i^t.
    for j = 1 to Q do
        for l ∈ I_i^{t,(k+1)} do
            Update every element l by Eq. (3.24).
```

After we obtain $\mathbf{S}_1^{t*}$ as $\mathbf{S}^t$, we then fix $\mathbf{S}^t$ to update $\mathbf{D}^t$.

3.3.2 Updating Common and Task-Specific Dictionaries

We update the dictionaries by fixing the sparse codes, thus, and the optimization problem becomes:

$$\min_{\hat{\mathbf{D}}^t, \bar{\mathbf{D}}^t} \mathcal{F}(\hat{\mathbf{D}}^t, \bar{\mathbf{D}}^t) = \frac{1}{2}||\mathbf{x}_i^t - [\hat{\mathbf{D}}^t, \bar{\mathbf{D}}^t]\mathbf{s}_i^t||_2^2 \tag{3.21}$$

We know the non-zero entries of $\mathbf{s}_i^{t,(k+1)}$ after we updating the sparse codes. The key insight of MMLC is that we just need to update the non-zero entries of the dictionaries but not all columns of the dictionaries, and it dramatically accelerates the optimization. When updating the i-th column and j-th row's entry of the dictionary $\mathbf{D}$, the gradient of $\mathbf{D}_{j,i}$ is set to be $\nabla \mathbf{D}_{j,i} = \mathbf{s}_i(\mathbf{D}_j^T\mathbf{s} - \mathbf{x}_j)$. If $\mathbf{s}_i = 0$, the gradient would be zero. We therefore do not need to update the $\mathbf{D}_j$. The learning rate is set to be an approximation of $1/\mathbf{H}_t^{k+1}$, which is updated by the sparse codes $\mathbf{s}_i^{t,(k+1)}$ in

k-th iteration. We first update the Hessian matrix $\mathbf{H}_t^{k+1}$ by:

$$\mathbf{H}_t^{k+1} = \mathbf{H}_t^k + \mathbf{s}_i^{t,(k+1)} \mathbf{s}_i^{(t,(k+1))T}. \tag{3.22}$$

One step SGD is performed to update the dictionaries: $\hat{\mathbf{D}}_t^{k+1}$ and $\bar{\mathbf{D}}_t^{k+1}$. We use a vector $\mathbf{R}$ to store the information $\mathbf{Dz} - \mathbf{x}$ in order to speed up the computation.

$$\mathbf{R} = \Omega([\hat{\mathbf{D}}^{t,(k)}, \bar{\mathbf{D}}^{t,(k)}], \mathbf{s}_i^{t,(k+1)}, \mathbb{I}_i^{t,(k+1)}) - \mathbf{x}_i^t. \tag{3.23}$$

Here, $\mathbf{R} = \tau([\hat{\mathbf{D}}^{(k-1)}, \bar{\mathbf{D}}^{t,(k-1)}], \mathbf{S}^{t,(k)}) - \mathbf{X}^t$, where $\tau(\mathbf{A}, \mathbf{B})$ is a matrix multiplication function and $\tau(\cdot) = \mathbf{AB}$. The procedure of learning the l-th column and j-th row of dictionaries takes the form of

$$[\hat{\mathbf{D}}_t^{k+1}, \bar{\mathbf{D}}_t^{k+1}]_{j,l} = [\hat{\mathbf{D}}^{t,(k)}, \bar{\mathbf{D}}^{t,(k)}]_{j,l} - \frac{1}{\mathbf{H}_t^{k+1}(l,l)} \mathbf{s}_{i,l}^{t,(k+1)} \mathbf{R}_j, \tag{3.24}$$

where l is the non-zero entry stored in $\mathbb{I}_i^{t,(k+1)}$. We let the learning rate be the inverse of the diagonal element of the Hessian matrix as $1/\mathbf{H}_t^{k+1}(l,l)$ for the l-th column of the dictionary.

It is important to normalize the dictionaries $\hat{\mathbf{D}}^{t,(k+1)}$ and $\bar{\mathbf{D}}^{t,(k+1)}$ after updating them because of $\mathbf{D}_t \in \Psi_t$ in equation (Eq. (3.21)). Since the dictionaries updating procedure only occurs at non-zero entries, we perform the normalization on the the corresponding columns of $\mathbf{s}_i^{t,(k+1)}$. The step of utilizing non-zero entries from $\mathbb{I}_i^{t,(k+1)}$ accelerates the whole learning process. We summarized the updating rules of dictionaries into Algorithm 7.

3.3.3 Updating Resemblance Term

After we update $\mathbf{D}^t$, we finally calculate $w_{p,q}$, and update the fourth term of Eq. (3.5) at the end of k-th epoch. We update the inherent resemblant knowledge term with the iterative soft-thresholding (Bredies and Lorenz, 2008). We first calculate the

gradient $\mathbf{g}$ based on Eq. (3.25), and then update the model $\mathbf{S}^{t,(k)}$ based on $\mathbf{g}$. The calculation of $\mathbf{g}$ and $\mathbf{S}^{t,(k)}$ follows the equations:

$$\begin{aligned} \mathbf{g} &= \frac{1}{\gamma}\mathbf{D}^t\mathbf{X}^t + [\mathbf{I} - \frac{1}{\gamma}(\mathbf{D}^{tT}\mathbf{D}^t + w_{p,q}\lambda_3\mathbf{I})]\mathbf{S}^{t,(k-1)}, \\ \mathbf{S}^{t,(k)} &= \Omega_{\lambda_3}(\mathbf{g} + w_{p,q}\frac{\lambda_3}{\gamma}\mathbf{D}^t), \end{aligned} \tag{3.25}$$

where γ is a non-negative parameter and Ω_{λ_3} is the soft-thresholding operator. Details of updating rules of MMLC updating rules can be found in Algorithm 5.

The convergence of MMLC algorithm is reached when the error of the objective function is below a threshold $\epsilon = 10^{-3}$ and the SVD of $\mathbf{S}$ can be computed efficiently with time complexity $O(mnl)$, where $l < min(m, n)$ is its rank. It is worth noting that the overall computational complexity of MMLC is $O(m^3 + \epsilon^{-0.5}mn + m^2n)$ when the number of IALM iterations is $O(\epsilon^{-0.5})$. This is much faster than the complexity of conventional method $O(m^3 + m^2n + mn^2)$.

Table 3.1: The Prediction Results of MMSE on Whole Dataset.

Methods	wR	M12	M18	M24	M36	M48
Lasso	0.40±0.09	4.04±0.77	3.46±0.97	5.53±0.86	4.39±0.74	4.73±1.49
Ridge	0.41±0.07	4.26±0.56	3.56±0.93	5.05±0.54	4.21±0.47	3.62±0.91
L21	0.57±0.01	3.32±0.63	4.75±0.75	4.64±0.88	4.08±1.01	3.11±1.05
ODL-L	0.63±0.08	2.99±0.63	**2.88±0.68**	4.29±0.84	3.62±1.45	2.93±1.07
TGL	0.70±0.05	2.73±0.72	4.00±1.31	4.00±0.64	3.19±1.38	2.60±1.42
MSMT	0.73±0.02	2.61±0.55	3.37±1.01	3.66±0.78	2.73±1.09	2.52±1.20
MMLC	**0.75±0.02**	**2.55±0.23**	2.99±0.89	**3.38±0.76**	**2.65±0.79**	**2.32±1.02**

Table 3.2: The Prediction Results of ADAS-cog on Whole Dataset.

Methods	wR	M12	M18	M24	M36	M48
Lasso	0.49±0.05	6.81±1.03	6.87±0.74	7.62±0.87	8.08±1.39	6.55±1.34
Ridge	0.46±0.07	7.68±0.96	6.89±1.69	7.84±1.54	8.59±0.62	6.64±1.58
L21	0.53±0.07	6.40±0.51	6.95±0.88	8.07±0.67	8.00±1.04	5.92±0.60
ODL-L	0.53±0.05	5.65±0.73	4.97±0.67	7.30±0.77	7.25±0.69	5.56±1.22
TGL	0.72±0.04	5.52±1.15	5.70±0.53	6.85±1.06	**6.36±1.22**	5.73±0.61
MSMT	0.77±0.02	5.18±0.88	**4.64±1.12**	6.76±1.35	6.78±1.54	5.27±1.76
MMLC	**0.80±0.04**	**5.17±0.95**	4.87±0.99	**6.66±0.65**	**6.37±1.23**	**5.16±1.31**

3.4 Experiments

3.4.1 Data and Experimental Settings

In this work, we study the performance of MMLC on the entire *ADNI-1 cohort*. We use structural MR images coming from seven different time points: baseline, 6-, 12-, 18-, 24-, 36- and 48-month.837, 733, 728, 326, 641, 454 and 251 are the sample sizes corresponding to seven time points, respectively. Thus, we learn a total of 3970 images and the responses are the Mini Mental State Examination (MMSE) and Alzheimer's Disease Assessment Scale cognitive subscale (ADAS-Cog) score. In addition, we remove 23 subjects who do not have MMASE and ADAS-cog information at baseline in this work.

Surface features

We use hippocampal surface multivariate morphometry statistics (MMS) Wang *et al.* (2011b) (Fig. 3.1 (c)) as our learning features. The original input data are the three-dimensional (3D) T1-weighted images (Fig. 3.1 (a)) from ADNI dataset. We first use FIRST(`https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FIRST`) to segment the original data and obtain the hippocampus substructure (Fig. 3.1 (b)). We then adopt the surface fluid registration Shi *et al.* (2013) to obtain surface geometric features for automated surface registration. Following that, a set of vertex-wise hippocampal MMS features are computed as Wang *et al.* (2011b). They consist of surface multivariate tensor-based morphometry (mTBM) and radial distance (RD). mTBM describes the surface deformation along the surface tangent plane while RD reflects surface differences along the surface normal directions. MMS features consist 4×1 vectors on each vertex of 15000 vertices on every hippocampal surface (each subject has two hippocampal surfaces). We select 1102 patches of size 10×10 on each hippocampal

surface mesh and each patch dimension is 400. We use the baseline and 6-month imaging data as training data and predict 12-month to 48-month clinical scores.

MMLC settings

The model was trained on an Intel(R) Core(TM) i7-6700 K CPU with 4.0GHz processors, 64 GB of globally addressable memory and a single Nvidia TITAN X GPU. The source code of MMLC are available at `http://gsl.lab.asu.edu/software/mmlc`. In the stage one, $\lambda_1 = 0.1$, $\lambda_2 = 10^{-2}$, $\lambda_3 = 10^{-3}$, $\mu_1 = 10$, $\mu_2 = 1$ and $\gamma = 1$, $\phi = 10$, the parameters were selected by cross-validation results on the training data. Also, we selected 10 epochs with a batch size of 1 and 3 iterations of CCD. In the Stage two, cross-validation is used to select model parameters ξ (between 10^{-3} and 10^3). In all experiments, we used 1000 atoms for the dictionary and 500:500 split atoms as the size of common and task-specific dictionaries (Sec. 3.4.2). When the sparse features were learned, Max-Pooling was used to generate features for annotation and finally we got a 1000-dimensional feature vector for each subject.

Evaluation method

In order to evaluate the model, we randomly split the data into training and testing sets using a 9:1 ratio to avoid data bias and report the mean and standard deviation based on 50 different splits of data. We evaluate the overall regression performance using weighted correlation coefficient (CC) and root mean square error (rMSE) for task-specific regression performance measures. The two measures are defined as $CC(\mathbf{Y}, \hat{\mathbf{Y}}) = \sum_{t=1}^{T} Corr(\mathbf{Y}^t, \hat{\mathbf{Y}}^t) n^t / \sum_{t=1}^{T} n^t$, $rMSE(\mathbf{Y}^t, \hat{\mathbf{Y}}^t) = \sqrt{||\mathbf{Y}^t - \hat{\mathbf{Y}}^t||_2^2 / n^t}$. For CC, $\mathbf{Y}^t$ is the ground truth of target of task t and $\hat{\mathbf{Y}}^t$ is the corresponding predicted value, $Corr$ is the correlation coefficient between two vectors and n^t is the number of subjects of task t. rMSE is computed for each task t, $\mathbf{Y}^t$ is the ground

truth of the target responses and $\hat{\mathbf{Y}}^t$ is the corresponding prediction. The smaller rMSE, the bigger wR mean the better results.

Comparison methods

We compare the proposed algorithm MMLC with six methods: 1) single-task regression method: LASSO Tibshirani (1996a) and Ridge Hoerl and Kennard (1970a). 2) multi-task regression: multi-task regression with $\ell_{2,1}$ norm regularization Liu *et al.* (2009b) (L21) and temporal group Lasso based multi-task progression model Zhou *et al.* (2012) (TGL). 3) sparse coding based method: single-task sparse coding followed by Lasso Zhang *et al.* (2016a) (STSC), Muilti-source Multi-target dictionary learning followed by Lasso regression Zhang *et al.* (2017c) (MTSC).

3.4.2 Experimental Results

The atoms of common and task-specific dictionaries

In Stage one of MMLC, the common dictionary is assumed to be shared by different tasks. It is necessary to evaluate what is an appropriate size of such common dictionary. Therefore, we set the dictionary size to be 1000 and partitioned the dictionary by different proportions: 125:875, 250:750,500:500, 750:250 and 875:125, where the left number is the size of common dictionary while the right one is the size of individual dictionary for each task. Fig. 3.3 shows the results of rMSE of MMSE and ADAS-cog prediction. As it shows in Fig. 3.3, the rMSE of MMSE and ADAS-Cog are lowest when we split the dictionary by half and a half. It means the both of common and individual dictionaries are of equal importance during the multi-task learning. In all experiments, we use the split of 500:500 as the size of common and individual dictionaries, the dimension of each sparse code in MMLC is 1000.

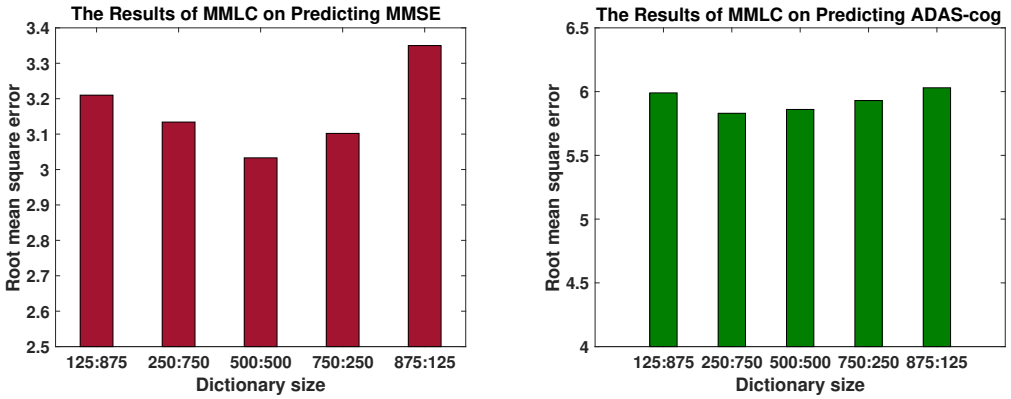


Figure 3.3: Comparison of rMSE Performance by Varying the Size of Common Dictionary.

Table 3.3: Time Comparisons of MMLC and STSC by Varying Dictionary Size on ADNI-I Dataset.

Dictionary Size	MMLC	STSC
500	1.74 hour	8.84 hour
1000	3.34 hour	21.95 hour
2000	6.93 hour	49.90 hour

The comparisons of time efficiency

We compare the efficiency of our proposed MMLC with STSC (Algorithm 4). In this experiment, we focus on the single batch size setting, that is, we process one image patch in each iteration. We vary the dictionary size as: 500, 1000 and 2000. For MMLC, the ratio between the common dictionary and the individual parts is 1:1. We report the results on ADNI-I cohort in Table 3.3. We observe that the proposed MMLC use less time than STSC. When the size of dictionary are increasing, MMLC is more efficient and has a higher speedup compared to STSC.

Comparison results on MMSE and ADAS-cog

We report the comparison results of MMLC and other methods of MMSE and ADAS-cog with ADNI-1 cohort in Table 3.1 and Table 3.2, respectively. In Table 3.1, the

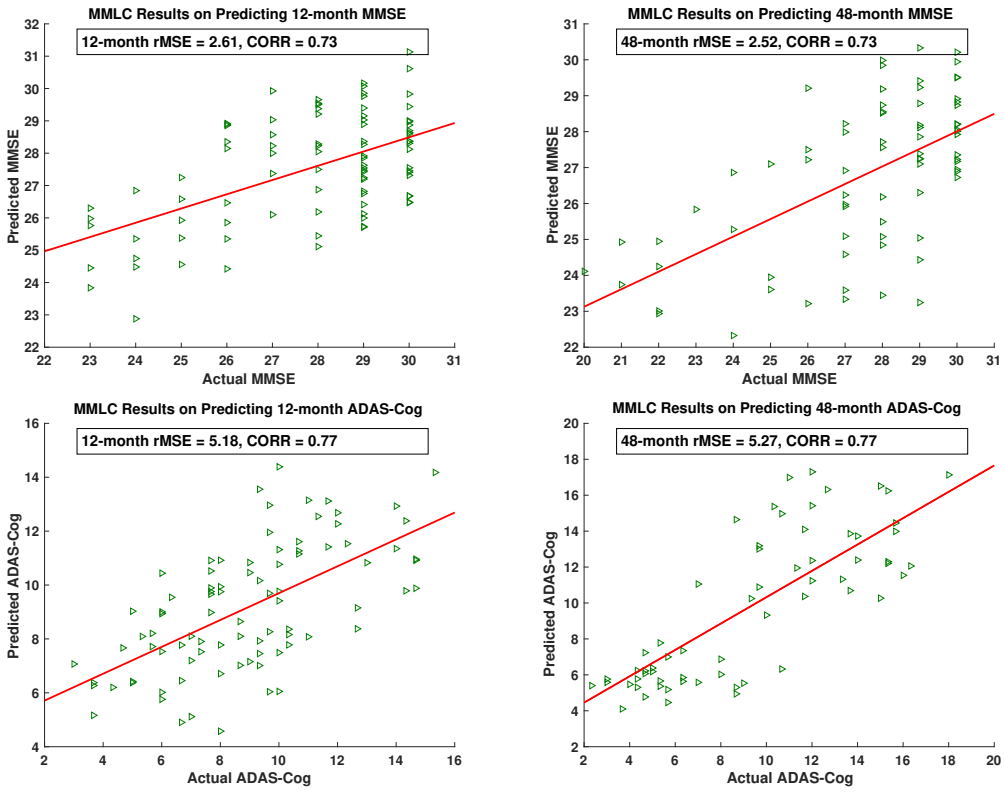


Figure 3.4: Scatter Plots of Actual MMSE and ADAS-Cog Versus Predicted Values on M12 and M48 by Using MMLC.

proposed MMLC outperforms linear regression methods in terms of both rMSE and correlation coefficient wR on four different time points. The results of Lasso and Ridge are very close while sparse coding methods are superior to them. For sparse coding methods, we observe that MTSC obtains lower rMSE and higher correlation results than STSC since MTSC considers the correlation between different time slots and the task-specific relationship. STSC has lower rMSE than MMLC on M18 because 18-month data is significantly less than other time points and SC has its bias on that point. We also notice that the proposed MMLC further improved the result of MTSC since we consider the low-rankness of the sparse codes and the resemblant knowledge in longitudinal dataset. Note that we significantly improve the rMSE results for later time points. A possible reason is that the baseline images has less correlation with later time points images and MTSC treats each time point equally.

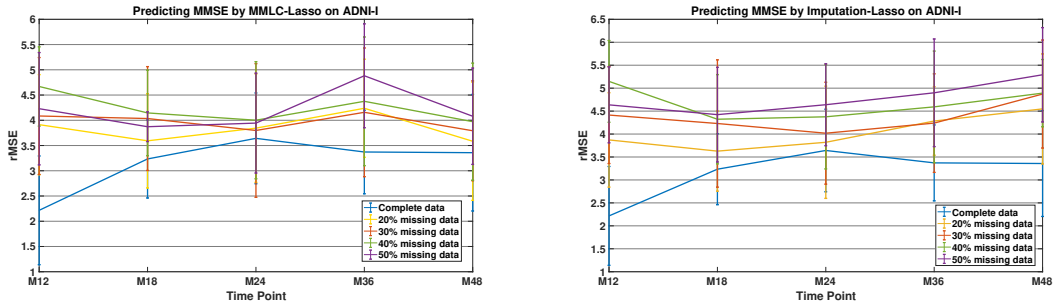


Figure 3.5: The rMSE Results of MMSE with Different Amount Missing Data by MMLC-Lasso and Imputation-Lasso, respectively.

In Table. 3.2, we can observe that the best performance of predicting scores of ADAS-Cog is achieved by MMLC for four time points. Comparing with L21, after MMLC dealing with missing label, the results are more linear, reasonable and accurate. Due to the dimension of M36 and M48 is too small, it is hard to learn a complete model. TGL also considers the issue of missing labels, however, MMLC achieves the better results because MMLC incorporates multiple sources data and uses common and individual dictionaries. This shows our method is more efficient about dealing with incomplete data.

We show the scatter plots for the predicted values versus the actual values for MMSE and ADAS-Cog on the M12 and M48 in Fig. 3.4. In the scatter plots, we see the predicted values and actual clinical scores have a high correlation. The scatter plots show that the prediction performance for ADAS-Cog is better than that of MMSE.

Ablation study on different amount of missing data

Furthermore, we study whether MMLC helps improve incomplete data results by varying different amount missing data. We start with a total of 122 subjects, which have complete MMSE value at all seven time points. We then randomly removed 20%, 30%, 40% and 50% target values during training. We perform our algorithm

MMLC to the complete data and different amount incomplete data. For comparison purpose, we apply the imputation approach Ito *et al.* (2010) to complete the missing data which uses neighboring time point data to approximate the missing value. For the experimental settings, we follow those of Sec. 3.4.1. Fig. 3.5 shows the rMSE results with different amount of missing data. The results show that compared with the imputation method Ito *et al.* (2010), our approach has better results that are close to the performance with the complete data.

3.5 Summary

In this chapter, I propose a novel multi-task sparse coding framework together with an efficient numerical scheme (MMLC). The experimental results clearly show MMLC offers a unique perspective on prognosis with longitudinal data. In the next chapter, I refine MMLC by considering a design of hierarchical model and adaptive large natural labeled data to the limited amount medical image data to further improve the prediction power and overcome the data scarcity.

Chapter 4

DEEP NATURAL DOMAIN ADAPTATION MULTI-ROIS LEARNING

4.1 Introduction

Recently, Convolutional Neural Networks (CNN) have been shown to be capable of learning the hierarchical structure of features extracted from real-world images and have been successfully applied to a variety of applications (Krizhevsky *et al.*, 2012). Feature learning with deep models typically requires a large amount of training data. Thus, feature learning for domains with scarce data is not feasible. Therefore, a key challenge in applying CNN to solving biological problems is that the available labeled training samples are insufficient. Transfer learning (Zhang *et al.*, 2015) is one of the approaches to address this problem and help feature learning in the data-scarce target domain by transferring knowledge from the data-rich source domain. However, it is still challenging to transfer the knowledge learned from natural images to the brain image analysis since the way that the output from the final layer is handled may not necessarily be the best as it contains more dataset-specific features. In this study, we aim to explore whether the nice domain adaptive property of CNN can be help apply CNN to general biological image research by empirical hypothesis testing.

In practice, even when we are able to successfully adaptive knowledge from some large amounts of natural imaging domain to brain imaging domain, employing transfer learning with a deep model on multiple brain region of interests (ROIs) data is still challenging. For example, in the study of learning the feature expression of *brain image*, it always associated with multiple promising anatomical regions. Such image representation is not suitable for formulating as a global image since it associates

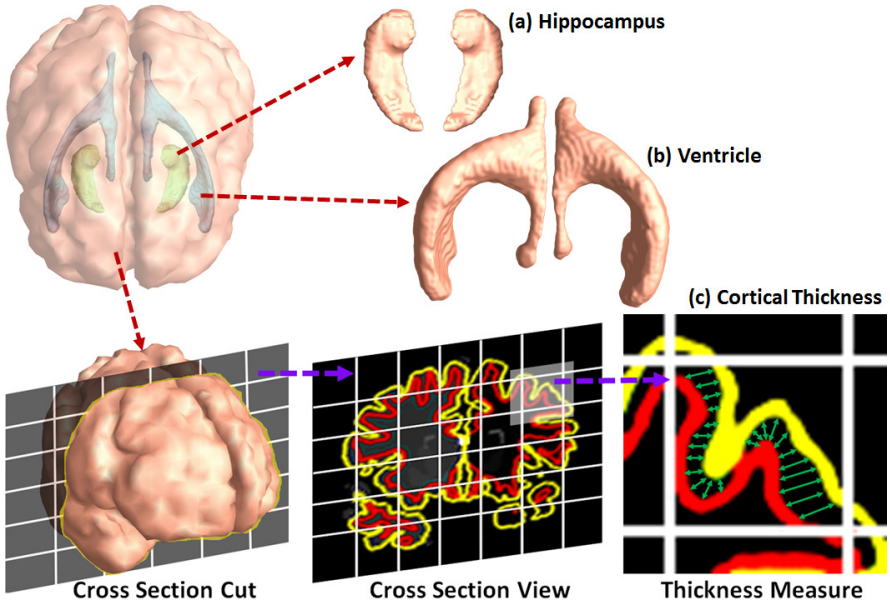


Figure 4.1: This Figure Shows Three Promising Anatomical ROIs in Brain Structural MR Images Used for Clinical Diagnosis of Alzheimer's Disease.

with several parts of local structural features. Fig. 4.1 depicts three most promising imaging Region of Interests (ROIs) associated with brain image analysis (Frisoni *et al.*, 2010a). A general pipeline for extracting local visual features is selecting local patches and combining the computed local features. However, MR images are 3D image and contain lots of spatial information, instead of selecting local patches, we need to capture the structural information. In Fig. 4.1, we can see that the hippocampus is connected to ventricle and thickness is along whole brain structure. Therefore, we argue that learning multiple associated ROIs as long as their spatial information can help we extract more meaningful features from the 3D brain images.

However, it is also necessary to reduce the feature dimension due to the large amounts of local patches of multiple ROIs from the limited number of subjects. Dictionary learning (Mairal *et al.*, 2009) has been proposed to use a small number of basis vectors termed dictionary to represent local features effectively and concisely and help image content analysis. However, most existing works on dictionary learning

focus on the prediction of a single time point target (Zhang *et al.*, 2016c) or multiple time point targets with single ROI data (Zhang *et al.*, 2017c). Here, we propose a novel approach that employs dictionary learning to identify important and concise features from multiple ROIs by adaptive natural image domains knowledge to brain image domain then predict multiple future time points clinical scores. The proposed Deep natural Domain Adaptation Multi-ROIs Learning (DDAML) is generic and can be applied to longitudinal features as well. Our work is expected to improve the performance of computer aided diagnosis and prognosis.

This work has three major contributions. First, we empirically demonstrate the feasibility of a direct domain adaptive learning from natural imaging domain (ImageNet) to brain imaging research. Second, we propose a multi-ROIs dictionary learning framework to reduce feature dimensions while considering the variance of features from different ROIs simultaneously and utilize shared and individual dictionary to encode both consistent and changing imaging features. Third, we test our hypothesis on multiple ROIs and the proposed DDAML outperforms three self-comparing methods and five other methods and is able to boost the performance of diagnoses ranging from cognitively unimpaired to AD.

4.2 Hypotheses

In this section, we are aiming to check on whether natural image domain can adaptive on brain image domain. Suppose the samples from two different data domains: source domain $\{\mathbf{X}_s\}$ and target domain $\{\mathbf{X}_t\}$ and their transformed versions can be view as $f_1(\mathbf{X}_s)$ and $f_2(\mathbf{X}_t)$, where the $f_1(\cdot)$ and $f_2(\cdot)$ are represent the objective function of source domain and target domain, respectively. We use idea of the hypothesis test to do an empirical experiments on verifying our hypothesis. Thus, the two hypotheses will be compared are listed as follows:

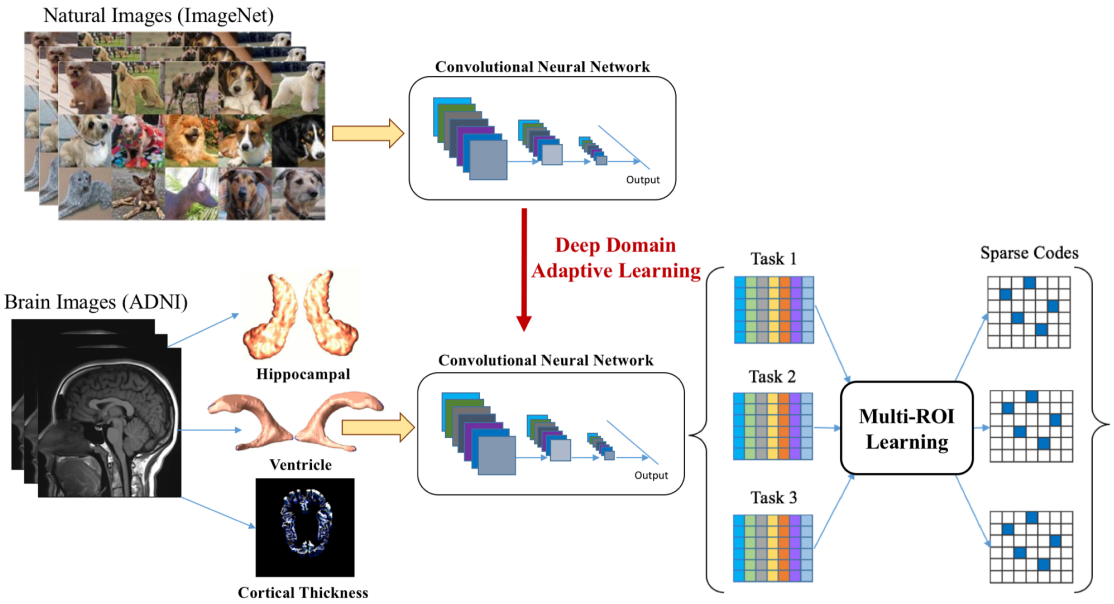


Figure 4.2: The Pipeline of Deep Natural Domain Adaptation Multi-ROIs Learning (DDAML). Adaptive the Knowledge from Natural Images to Brain Images by Convolutional Neural Network and Multi-ROIs Learning Integrates Three Types of Anatomical Features and Predicts Individual Clinical Scores by Concatenating Multiple Sparse Codes Features.

- $\mathbf{H_0}$: Given λ_1, there does not exist parameters $\lambda_2 = \lambda_1$, such that $f_2(\mathbf{X}_t)$ has a lower loss error.

- $\mathbf{H_1}$: Given λ_1, there exist parameters $\lambda_2 = \lambda_1$, such that $f_2(\mathbf{X}_t)$ has a lower loss error.

There are two possible outcomes either reject $\mathbf{H_0}$ or accept $\mathbf{H_0}$. It is straightforward to see that if we can find such parameters $\lambda_2 = \lambda_1$ that the loss error of $f_2(\mathbf{X}_t)$ can be decreased, then we will accept $\mathbf{H_1}$ and reject $\mathbf{H_0}$.

Our first goal here is to explore whether we accept $\mathbf{H_0}$. Let us assume we cannot find the λ_2 from the target domain (brain images). We design a fitted model on the following three testing tasks, 1) training from scratch on brain image data 2) using the same parameter $\lambda_2 = \lambda_1$ from natural images on brain image by the same training network 3) using the same parameter $\lambda_2 = \lambda_1$ from natural images fine-tuning on brain images (partial $\lambda_1 = \lambda_2$). Specifically, we use ImageNet (Deng *et al.*, 2009) data as

our natural images source domain data, containing millions of labeled natural images with thousands of categories to obtain initial parameters and subsequently generate the features on the longitudinal data for each task. We use the Alzheimer's Disease Neuroimaging Initiative (ADNI) (Weiner *et al.*, 2013) dataset as our target domain data, which is an ongoing, longitudinal, multicenter study designed to develop clinical, imaging, genetic, and biochemical biomarkers for the early detection and tracking of Alzheimers disease (AD). In the experiments, we apply Alexnet (Krizhevsky *et al.*, 2012), which contains seven layers, including convolutional layers with fixed filter sizes and different numbers of feature maps. We employ rectified non-linearity, max-pooling on each layer in our model. After we pretrain the CNN model on the ImageNet dataset, we remove the last fully-connected layer (this layer's outputs are the 1000 class scores for ImageNet). Finally, we treat the rest of the CNN as a fixed feature extractor for the geometry mesh extracted from ADNI. For the validation purpose, we fine-tune the pretrain AlexNet model of ImageNet on the geometry mesh extracted from ADNI as our third testing tasks.

4.3 Methods

The entire pipeline of our method is illustrated in Fig. 4.2. To be specific, after we transferred the knowledge from natural domain by CNN, we then explore the local feature from multi-ROIs to boost the global brain geometry mesh data. We further propose DDAML to generate the sparse features and dictionaries from the deep ROIs features. Additionally, we utilized shared and individual dictionaries to encode both consistent and changing imaging features from multiple ROIs. In the end, we employed the sparse features to perform the Lasso Tibshirani (1996a) and predict the future AD progression.

Given features from R different ROIs from the target domain $\{\mathbf{X}_t\} : \{\mathbf{X}_1, ..., \mathbf{X}_R\}$, our objective is to learn a set of sparse codes $\{\mathbf{Z}_1, ..., \mathbf{Z}_R\}$ for each task where $\mathbf{X}_r \in \mathbb{R}^{p \times n_r}$, $\mathbf{Z}_r \in \mathbb{R}^{l_r \times n_r}$ and $r = 1, ..., R$. n_r is the number of subjects for $\mathbf{X}_r$ and l_r is the dimension of each sparse code in $\mathbf{Z}_r$. When employing the online dictionary learning Mairal *et al.* (2009) to learn the sparse codes $\mathbf{Z}_r$ by $\mathbf{X}_r$ individually, we obtain a set of dictionary $\{\mathbf{D}_1, ..., \mathbf{D}_R\}$ but there is no correlation between learnt dictionaries. Another solution is to construct the features $\{\mathbf{X}_1, ..., \mathbf{X}_R\}$ into one matrix $\mathbf{X}$ to obtain the dictionary $\mathbf{D}$. However, if there is no latent common information shared by the same subject among different ROIs, only one dictionary $\mathbf{D}$ is not enough to show the correlation or variation among features from different ROIs. Such fact is supposed to be easily revealed in the variance of dictionary atoms and the sparsity of their corresponding sparse code matrices. To address this challenge, we integrate the idea of multi-task learning into the online dictionary learning method, which advantages from solving the cases that the size of the input data might be too large (sample size n_r up to 2,867,562) to fit into memory or the input data comes in a form of a stream.

For the subjects' feature matrix $\mathbf{X}_r$ of a particular task, DDAML learns a dictionary $\mathbf{D}_r$ and sparse codes $\mathbf{Z}_r$. $\mathbf{D}_r$ is composed of two parts: $\mathbf{D}_r = [\hat{\mathbf{D}}_r, \bar{\mathbf{D}}_r]$ where $\hat{\mathbf{D}}_r \in \mathbb{R}^{p \times \hat{l}}$, $\bar{\mathbf{D}}_r \in \mathbb{R}^{p \times \bar{l}_r}$ and $\hat{l} + \bar{l}_r = l_r$. $\hat{\mathbf{D}}_r$ is the same among all the learnt dictionaries while $\bar{\mathbf{D}}_r$ is different from each other and only learnt from the corresponding subjects' feature matrix $\mathbf{x}_r$. Therefore, the objective function of DDAML can be formulated as follows:

$$\min \quad ||\mathbf{L}(f_2(\mathbf{x}_t))||_2^2 : \lambda_2 = \lambda_1$$

$$\min_{\mathbf{D}_1, \cdots, \mathbf{D}_R, \mathbf{Z}_1, \cdots, \mathbf{Z}_R} \sum_{r=1}^{R} \frac{1}{2} ||\mathbf{x}_r - [\hat{\mathbf{D}}_r, \bar{\mathbf{D}}_r]\mathbf{Z}_r||_F^2 + \lambda \sum_{r=1}^{R} ||\mathbf{Z}_r||_1$$

$$s.t. \hat{\mathbf{D}}_1 = \cdots = \hat{\mathbf{D}}_R, \mathbf{D}_r \in \Psi_r,$$

where $\Psi_r = \{\mathbf{D}_r \in \mathbb{R}^{p\times l_r} : \forall j \in 1, ..., l_r, ||[\mathbf{D}_r]_j||_2 \leq 1\}$ and $[\mathbf{D}_r]_j$ is the jth column of $\mathbf{D}_r$, λ_1 and λ_2 are the parameters in Sec. 4.2 and $\mathbf{L}(\cdot)$ donates the loss function of the prediction task.

To solve the optimization problem, we employ stochastic coordinate coding Lin *et al.* (2014). The way we initialize $\hat{\mathbf{D}}_r$ is to randomly select $\hat{l}$ subjects' feature from features' matrices across different brain ROIs $\{\mathbf{X}_1, \cdots, \mathbf{X}_R\}$ to construct it. For the individual part of each dictionary, we randomly select $\bar{l}$ subjects' feature from the corresponding matrix $\mathbf{X}_r$ to construct $\bar{\mathbf{D}}_r$. After initializing dictionary $\mathbf{D}_r$ for each task, we set all the sparse code $\mathbf{Z}_r$ to be zero at the beginning.

The key steps of DDAML are summarized in two step. 1) For each subject's feature $\mathbf{x}_r(i)$ extracted from $\mathbf{X}_r$, we learn the ith sparse code $\mathbf{z}_r^{k+1}(i)$ from $\mathbf{Z}_r$ by several steps of Cyclic Coordinate Descent (CCD) Canutescu and Dunbrack (2003). 2) We use learnt sparse codes $\mathbf{z}_r^{k+1}(i)$ to update the dictionary $\hat{\mathbf{D}}_r^{k+1}$ and $\bar{\mathbf{D}}_r^{k+1}$ by one step Stochastic Gradient Descent (SGD) Zhang (2004). Since $\mathbf{z}_r^{k+1}(i)$ is very sparse, we use the index set $\mathbf{I}_r^{k+1}(i)$ to record the location of non-zero entries in $\mathbf{z}_r^{k+1}(i)$ to accelerate the update of sparse codes and dictionaries. Φ represent the shared part of each dictionary $\mathbf{D}_r$ which is initialized by the random patch method and is updated in the end of kth epoch to ensure $\hat{\mathbf{D}}_r^{k+1}$ is the same among all the dictionaries.

4.4 Experiments

4.4.1 Data and Experimental Settings

We built a prediction model for multiple ROI geometry features using multiple task geometry surface features computed as Wang *et al.* (2011b). To train the CNN model, patches of size 50×50 are extracted from surface mesh structures. We implemented our CNN model using the Caffe toolbox (`http://caffe.berkeleyvision.org/`) and

the architecture of our CNN is AlexNet (Krizhevsky *et al.*, 2012). The network was trained on a Intel (R) Xeon (R) 48-core machine, with 2.50 GHZ processors, 256 GB of globally addressable memory and a single Nvidia Tesla K40 GPU. In the experimental setting of MROI, the sparsity $\lambda = 0.1$. Also, we selected 10 epochs with a batch size of 1 and 3 iterations of CCD in all experimental settings. After we get the sparse features, we used Max-Pooling (Boureau *et al.*, 2010) for further dimension reduction. Therefore, the feature dimension of each subject is a 1×2000 vector since $l = 2000, \hat{l} = 1000$ and $\bar{l} = 1000$. To predict future clinical scores, we used Lasso regression. For the parameter selection, 5-fold cross-validation is used to select model parameters in the training data (between 10^{-3} and 10^{3}). We used the same method for all comparison methods.

In this experiment, we utilized three structural measures of brain, which are hippocampus (Wang *et al.*, 2011b), lateral ventricle (Wang *et al.*, 2011b) and cortical thickness (Chung *et al.*, 2008b), from the ADNI baseline dataset ($N = 837$). In brief, for the hippocampal surface features, we used the same methods as (Zhang *et al.*, 2016c) and obtained a 120,000 dimensional features of the hippocampal surfaces while for the ventricular surface features we did the following. We segmented images of the lateral ventricles to build the ventricular structure surface models using a level-set based topology preserving method and computed surface registrations using the canonical holomorphic one-form segmentation method (Wang *et al.*, 2009). Finally, surface multivariate morphometry statistics (Wang *et al.*, 2011b) were computed and obtained as a 308,247 dimensional features of the ventricular surfaces for each subject. The cortical thickness was computed by FreeSurfer (`https://surfer.nmr.mgh.harvard.edu/`) which deforms the white surface to pial surface and measures deforming distance as the cortical thickness. The spherical parameter surface and weighted spherical harmonic representation (Chung *et al.*, 2008b) are

used to register pial surfaces across subjects, which means each subjects have the same dimension (161,800) cortical thickness. The image patch size is 50×50 and after preprocessing the data, we have 220968, 2867562, 1504926 image patches for multiple input tasks, respectively.

4.4.2 Experimental Results

Results of Natural Domain Adaptation Learning. Table 4.1 shows the results of our empirical hypothesis test, and we can obtain three information from it: 1) adaptive the natural images domain on brain images domain can improve the disease prediction results. 2) There exists parameters $\lambda_2 = \lambda_1$ to accept the alternate hypothesis. 3) Fine-tuning the pretrain model can improve the prediction results comparing with the first experiment but achieved less performance compared with fully adaptive all parameters from the source domain, which also give us evidence to reject the null hypothesis. Therefore, we reject $\mathbf{H}_0$ and accept $\mathbf{H}_1$ for our hypothesis test.

Results of DDAML. We compared the proposed model with three self-variate methods and five other methods. *DDAML* is our proposed pipeline; *DSDML* is CNN learned surface features on single ADNI domain with multi-ROIs learning followed by Lasso; *DSD* is CNN learned surface features on single domain, followed by Lasso; *MRL* is multi-ROIs learning followed by Lasso without deep transfer learning; *SRL* is single-ROI online dictionary learning (Mairal *et al.*, 2009) followed by Lasso; *cFSGL* is a multi-task algorithm called convex fused sparse group Lasso (Zhou *et al.*, 2013); *L21* is a multi-task algorithm called $L_{2,1}$ norm regularization with least square loss (Argyriou *et al.*, 2008); *Lasso* is a single task method Lasso regression (Tibshirani, 1996a); *Ridge* is a single task method Ridge regression (Hoerl and Kennard, 1970a).

We first form the final baseline data by concatenating three ROIs sparse features. Then, we individually predict 6-month, 12-month and 24-month MMSE and ADAS-cog scores. The prediction results are reported in Table 4.2 and Table 4.3. We can observe that the performance of predicting 6-month, 12-month and 24-month scores of MMSE and ADAS-Cog are improved by DDAML, DSDML and MRL for all three time points and DDAML achieved the best result among these three methods, which shows the deep natural-domain learning can help improve the MRL. We can also observe the deep learning method extract the better features compare with directly apply MRL and MRL obtained a lower rMSE result than SR since we consider the correlation between different tasks. We can also notice that the significant improvement of the proposed DDAML and MRL for later time points (M12, M24). This may be due to the data sparseness in later time points, as the proposed sparsity-inducing models are expected to achieve better prediction performance. Also, the improvement of ADAS-cog is more significant than MMSE.

Table 4.1: The Results of Natural Domain Adaptive Learning on MMSE and ADAS-cog.

Method	MMSE		ADAS	
	nMSE	wR	nMSE	wR
Only target domain	0.311±0.051	0.681±0.091	0.802±0.059	0.712±0.058
$\lambda_1 = \lambda_2$	**0.274±0.051**	**0.751±0.083**	**0.762±0.012**	**0.862±0.045**
Partial $\lambda_1 = \lambda_2$	0.291±0.074	0.683±0.091	0.748±0.067	0.749±0.368

Table 4.2: The MMSE Results of 6-month, 12-month and 24-month.

MMSE	nMSE	wR	M06	M12	M24
DDAML	**0.274±0.051**	**0.751±0.083**	**2.198±0.062**	**2.211±0.459**	**2.290±0.601**
DSDML	0.308±0.097	0.719±0.075	2.330±0.079	2.480±0.342	2.799±0.645
DSD	0.311±0.051	0.681±0.091	2.218±0.062	2.396±0.250	2.591±0.420
MRL	0.308±0.058	0.654±0.036	2.451±0.357	2.566±0.560	2.859±0.494
SRL	0.337±0.112	0.692±0.074	2.578±0.319	2.954±0.746	3.706±0.711
cFSGL	0.312±0.037	0.726±0.066	2.424±0.315	2.691±0.272	2.906±0.907
L21	0.281±0.032	0.572±0.082	2.535±0.473	2.897±0.990	3.107±0.501
Lasso	0.302±0.078	0.423±0.073	2.659±0.804	2.904±0.658	3.335±0.692
Ridge	0.299±0.101	0.449±0.091	2.766±0.776	3.001±0.280	3.621±0.893

Table 4.3: The ADAS-cog Results of 6-month, 12-month and 24-month.

ADAS-cog	nMSE	wR	M06	M12	M24
DDAML	**0.762±0.012**	**0.862±0.045**	**4.322±0.269**	**4.930±0.192**	**5.521±0.816**
DSDML	0.797±0.094	0.837±0.034	4.432±0.765	5.022 ±0.584	5.898±1.022
DSD	0.802±0.059	0.712±0.058	5.521±0.712	5.913±0.213	6.012±0.941
MRL	0.792±0.039	0.837±0.045	4.506±0.452	5.124±0.689	5.835±1.042
SRL	0.828±0.079	0.681±0.052	5.080±0.589	5.860±0.608	6.179±1.001
cFSGL	0.795±0.052	0.836±0.031	4.451±0.340	5.230±0.589	6.249±0.996
L21	0.811±0.080	0.554±0.062	4.476±0.931	5.453±0.392	6.279±1.232
Lasso	0.809±0.110	0.518±0.080	5.295±0.763	5.799±1.001	6.729±0.705
Ridge	0.819±0.108	0.497±0.071	5.534±0.542	5.907±0.885	6.543±0.844

4.5 Summary

In this chapter, we proposed a deep natural domain adaptation multi-ROIs learning (DDAML) algorithm which transfers knowledge from ImageNet to brain ROIs for predicting the AD clinical score. Our proposed model is generic and may also be applied to consolidate imaging information from any longitudinal dataset. In the next chapter, I comprehensively capture temporal-subject sparse features towards earlier and better discriminability of AD.

Chapter 5

TEMPORALLY ADAPTIVE-DYNAMIC SPARSE NETWORK

5.1 Introduction

Accurately predict disease progression is a big challenge because of the paramount difficulty of modeling disease association between the limited amount of brain images and specific clinical measures at multiple time points.

In recent years, various methods were proposed to address the above challenges. Zhou *et al.* (2013) considered a convex fused sparse group Lasso formulation to model disease progression, which successfully utilized the intrinsic relationships among multiple related tasks. Suk *et al.* (2016) proposed a deep learning-based sparse multi-task regression to jointly analyze the neuroimaging and clinical data in a prediction of the memory performance. Sparse coding (SC) (Jiang *et al.*, 2015b; Zhang *et al.*, 2016c) has been demonstrated a great success in using the sparse basis representation to extract local features effectively to help model disease progression. A multi-task sparse coding (Zhang *et al.*, 2017c) framework was proposed to predict clinical scores of multiple time points while neglecting the essential temporal information of the longitudinal data. Recently, deep learning methods (Wang *et al.*, 2018; Zhang *et al.*, 2017d; Zhang and Wang, 2019b) were adopted to address the temporal information in disease modeling, but it ignores the subject-level information which might further improve the predictive performance.

Despite the prosperity and progress achieved in the AD prediction, there are still several drawbacks appearing in the above methods. First, previous SC methods train multiple time points data simultaneously, thus ignore the inherent temporal structure

among these time points. Although recurrent neural network (RNN)-based methods consider such temporal information, it cannot capture the strong sparsity pattern in the longitudinal cohort data and may result in sub-optimum solutions. Second, it has been shown that feature extraction with the SC is consistent only under certain conditions (Meinshausen *et al.*, 2006). As a result, there is a need to develop an algorithm which regularizes the sparse codes consistency along the temporal longitudinal pattern. Third, the dictionary of SC only encodes patch-level atoms, which means a loss of information when the sparse features are well-aligned across multiple related patches. Learning the dictionary on the entire subject's patches could make use of all available information and reveal fundamental sparse features. To overcome the above limitations, we propose a novel extension of SC to the common recurrent neural network to model the AD progression. We name it Temporally Adaptive-Dynamic Sparse Network (TaDsNet, Fig. 5.1). It is a supervised SC scheme built into a RNN that 1) adaptively regularizes the sparse codes along the temporal longitudinal pattern of the dataset to improve the global sparse regularization structure and, 2) dynamically optimize the entire subject-level features over single selected image patches.

We summarize our most significant contributions into threefold. Firstly, we propose to adopt adaptive weights to regularize the sparse codes along longitudinal patterns of the features. Meanwhile, the adaptive structure makes it very powerful in modeling temporal sparsity patterns, especially for longitudinal data, and particularly useful in high-dimensional problems. Our approach can adaptively set various sparsities of sparse codes to minimize the errors. Secondly, we suggest that dictionary atoms should be learned on the entire subject to provide global high-level features. Thus our approach can dynamically mine the dictionary atoms to learn the subject-level features better than patch-level features. Thirdly, taking advantage of the RNN, we model the disease progression via feeding the longitudinal data into a time sequence

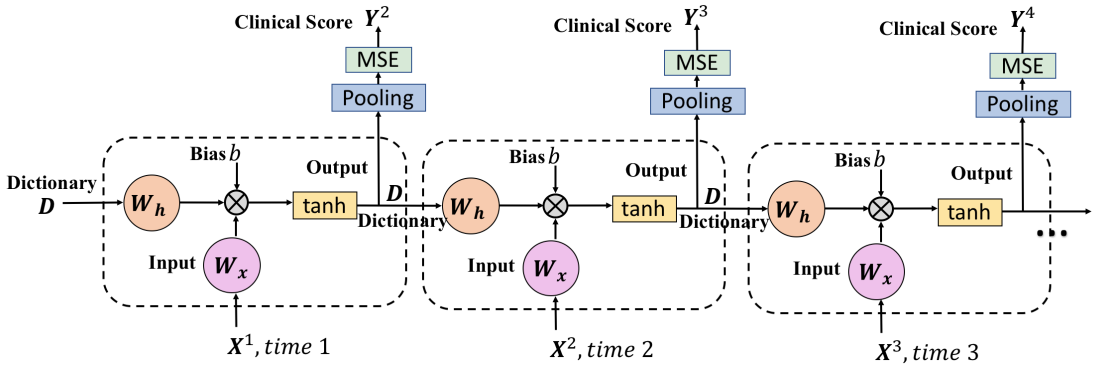


Figure 5.1: Illustrate the Architecture of the Proposed TaDsNet, which Learns the Input $\mathbf{X}^t$ in Time Sequence and Predict the Clinical Scores for Next Time Point $\mathbf{Y}^{t+1}$.

network. Different from previous methods, our approach is a supervised time-series sparse coding, which can fully leverage the temporal and clinical patterns derived from patients past visits. To the best of our knowledge, this is the first supervised network-based SC to model the AD progression. It adaptively adjusts the sparse codes and dynamically explores dictionary atoms on the entire subject-level in the RNN temporal learning mode. The experimental results demonstrate that TaDsNet achieves significant improvement in terms of both model performance and effectiveness compared with other related methods.

5.2 Methods

5.2.1 Problem Definition

Given an input matrix $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{m \times n}$, where each selected patch $\mathbf{x}_i \in \mathbb{R}^m$. Sparse coding is aiming to learn a dictionary $\mathbf{D} = [\mathbf{d}_1, \cdots, \mathbf{d}_m]^T \in \mathbb{R}^{m \times p}$ and a sparse code matrix $\mathbf{S} = [\mathbf{s}_1, \cdots, \mathbf{s}_n] \in \mathbb{R}^{p \times n}$. The original input matrix $\mathbf{X}$ is modeled by a sparse linear combination of $\mathbf{D}$ and $\mathbf{S}$ as $\mathbf{X} \approx \mathbf{D^T S}$. We can formulate the following optimization problem:

$$\min_{\mathbf{D}\in\Psi,\mathbf{S}} f(\mathbf{D},\mathbf{S}) = \frac{1}{2}||\mathbf{X}-\mathbf{D}\mathbf{S}||_F^2 + \lambda||\mathbf{S}||_{1,1}, \tag{5.1}$$

where $\Psi = \{\mathbf{D} \in \mathbb{R}^{p\times m} : \forall j \in 1,\cdots,p, ||\mathbf{d}_j||^2 \leq 1\}$, λ is the positive regularization parameter and $||\cdot||_F$ denotes the Frobenius norm of the matrix. Eq. (5.1) is a non-convex problem, however, it will become two convex problems, dictionary learning and sparse approximation, when we alternatively optimize $\mathbf{D}$ and $\mathbf{S}$.

Specifically, Eq. (5.1) reduces to the following quadratically constrained optimization problem by fixing sparse codes $\mathbf{S}$,

$$\min_{\mathbf{D}} \frac{1}{2}||\mathbf{X}-\mathbf{D}\mathbf{S}||_F^2, \quad s.t. ||\mathbf{d}_j||^2 \leq 1, j = 1,\cdots,p, \tag{5.2}$$

where Eq. (5.2) is the well-known ridge regression problem (Hoerl and Kennard, 1970b), which has a closed-form solution.

When it comes to update sparse codes $\mathbf{S}$ by fixing $\mathbf{D}$, Eq. (5.1) can be reduced to represent the input $\mathbf{X}$ by a sparse linear combination of $\mathbf{D}$ as follows,

$$\min_{\mathbf{s}_i} \sum_{i=1}^{n} (\frac{1}{2}||\mathbf{x}_i - \mathbf{D}\mathbf{s}_i||_2^2 + \lambda||\mathbf{s}_i||_1). \tag{5.3}$$

ISTA (Daubechies *et al.*, 2004) is usually used to solve Eq. (5.3). To be specific, ISTA updates the first term of Eq. (5.3) by gradient descent and the ℓ_1-norm is updated by hard thresholding. We summarize the mathematically update rule as follows:

$$\begin{aligned}
\mathbf{S}^{\kappa} &= sh_{(\eta\lambda)}(\mathbf{S}^{\kappa-1} - \lambda \bigtriangledown g(\mathbf{S}^{\kappa-1})), \\
&= sh_{(\eta\lambda)}(\mathbf{S}^{\kappa-1} - \lambda(\mathbf{D}^T(\mathbf{D}\mathbf{S}^{\kappa-1} - \mathbf{X}))), \\
&= sh_{(\eta\lambda)}(\mathbf{W}_h\mathbf{S}^{\kappa-1} + \mathbf{W}_x\mathbf{X}),
\end{aligned} \tag{5.4}$$

where $sh_{(\eta\lambda)}(\mathbf{S}) = sign(\mathbf{S})(|\mathbf{S}| - \eta\lambda)_+$ is the shrinkage function (Daubechies *et al.*, 2004), κ is the epoch, $\mathbf{W}_h = \mathbf{I} - \lambda\mathbf{D}^T\mathbf{D}$, and $\mathbf{W}_x = \lambda\mathbf{D}^T$. LISTA (Gregor and LeCun, 2010) is proposed to unfold the above Eq. (5.4) into a simple RNN as $\mathbf{W}_h$ and $\mathbf{W}_x$ are the hidden and inner weights of RNN unit, so we can learn $\mathbf{S}$ by RNN once we learned $\mathbf{D}$.

However, a problem exists in Eq. (5.1) is its inability to distinguish between correlated data. Unfortunately, longitudinal MR images are correlated among multiple time points and learn sparse features on individual image patch cannot promise the subject-level features. We suggest adaptively regularize the sparse codes to learn the temporal intrinsic features and atoms should be learned on the entirety subject's patches to provide global subject-level insights.

5.2.2 Temporally Adaptive-Dynamic Sparse Network

In this chapter, we propose TaDsNet, which can adaptively and dynamically learn the sparse features of Eq. (5.1). We use an initial perturbations (Zhang *et al.*, 2016d) on Eq. (5.1) to adaptively let all features can be selected by competing with each other.

$$\min_{\mathbf{D}\in\Psi,\mathbf{s}_i,\Omega}\sum_{i=1}^{n}(\frac{1}{2}||\mathbf{x}_i-\mathbf{D}\mathbf{s}_i||_2^2+||\Omega^{-\eta}\odot\mathbf{s}_i||_1), \tag{5.5}$$

where $\Omega\in\mathbb{R}^p$ denotes regularization weight, each element of $\mathbf{s}_i$ will receive different penalty instead of the same sparsity by λ as Eq. (5.3). In addition, $|\Omega\odot\mathbf{s}_i|_1=\sum_i\omega_i^{-\eta}|\mathbf{s}_i|$ and $\Omega=|\mathbf{s}_i^*|^{-\eta}$, where $\mathbf{s}_i^*$ is the ordinal least-square solution.

We then hypothesis that the set of patches of interest atoms may adapt their position across the whole subject. We introduce the shift operator (Hitziger *et al.*, 2013) Θ on atoms to modify Eq. (5.5). Given a set of shift operations Θ that contains only small shifts relative to the number of the patches, for every j there exists coefficient $s_{ij}\in\mathbb{R}$ and operator $\theta_{ij}\in\Theta$, such that $\mathbf{x}_j=\sum_i s_{ij}\theta_{ij}(\mathbf{d}_i)$. We therefore can formulate the temporally adaptive-dynamic sparse network as follows:

$$\begin{aligned}\min_{\mathbf{d}_i,\mathbf{s}_{ij},\theta_{ij}\in\Theta,\Omega}&\sum_{j=1}^{n}(\frac{1}{2}||\mathbf{x}_j-\sum_{i=1}^{p}s_{ij}\theta_{ij}(\mathbf{d}_i)||_2^2+||\Omega^{-\eta}\odot\mathbf{s}_j||_1),\\ &s.t.\ ||\mathbf{d}_i||_2=1.\end{aligned} \tag{5.6}$$

Optimization: For updating the dictionary $\mathbf{D}$, we use block coordinate descent (Tseng, 2001) for updating each atom $\mathbf{d}_k$,

$$\mathbf{d}_k = \arg\min_{\mathbf{d}_k} \frac{1}{2} \sum_{j=1}^{n} ||\mathbf{x}_j - \sum_{i=1}^{p} s_{ij}\theta_{ij}(\mathbf{d}_i)||_2^2, \ s.t. \ ||\mathbf{d}_k||_2 = 1. \tag{5.7}$$

Eq. (5.7) can be solved in two steps, the solution of the unconstrained problem is the differentiation followed normalization as follows:

$$\tilde{\mathbf{d}}_k = \sum_{j=1}^{n} s_{kj}\theta_{kj}^{-1}(\mathbf{x}_j - \sum_{i \neq k} s_{ij}\theta_{ij}(\mathbf{d}_i)), \quad \mathbf{d}_k = \frac{\tilde{\mathbf{d}}_k}{||\tilde{\mathbf{d}}_k||_2}. \tag{5.8}$$

In Eq. (5.8), $\theta\theta^T = I$. If the shift operator is a non-circular operators, the inverse θ_{kj}^{-1} needs to be replaced by the adjoint θ_{kj}^T and the rescaling function $\phi = (\sum_{j=1}^{n} s_{kj}^2 \theta_{kj}\theta_{kj}^T)^{-1}$ needs to be applied to the second update term in Eq. (5.8).

Now, we fix the dictionary $\mathbf{D}$ after we update it by Eq. (5.8) and we solve $\mathbf{S}$ (Eq. (5.6)) as solving a LASSO problem (Tibshirani, 1996b) with adaptive weights (Zhang *et al.*, 2016d) to regularize the correlation features along with the sparse codes. We then gradually shrink the solution by using stronger ℓ_1-penalties and fewer features remaining in the progressive shrinking and will go through the self-adjusting sequential stages before reaching the final optimal. Notice that such adaptive shrinking procedure does not result in a significant loss of performance.

We use an adaptive weight vector $\Omega = [\omega_1, \cdots, \omega_n]^T \in \mathbb{R}^p$ to regularize over different covariates, as

$$\min_{\Omega, \mathbf{s}_i} \sum_{i=1}^{n} (\frac{1}{2}||\mathbf{x}_i - \mathbf{D}\mathbf{s}_i||_2^2 + ||\Omega^{-\eta} \odot \mathbf{s}_i||_1), \quad s.t. \sum_i \Omega_i, \omega_i \geq 0, \tag{5.9}$$

where $||\Omega^{-\eta} \odot \mathbf{s}_i||_1 = \sum_{j=1}^{n} \Omega_j^{-\eta} \cdot |s_{ij}|$ and η is the shrinking factor. We alternatively optimize Ω and $\mathbf{S}$ as the following learning process.

Suppose we initialize $\Omega = \Omega_0$, we alternatively update Ω and $\mathbf{S}$ in Eq. (5.9) under this equality norm constraint until convergence. When we fix Ω, solving $\mathbf{S}$ can use LISTA (Gregor and LeCun, 2010) with Eq. (5.4). We then start the second stage of

iterations with an updated norm constraint $|\Omega| = \Omega_1$ after the initial stage, which imposes a stronger ℓ_1 penalty. Then we alternatively update Ω and $\mathbf{S}$ until the second stage ends. We keep strengthening the global ℓ_1-norm regularization stage by stage until the algorithm converges. We use τ to denote the index of each stage and $|\Omega| = \Omega_\tau$ is the compose of the iteration. Notice, when we fix $\mathbf{S}$ update Ω, the problem becomes the following constrained optimization problem:

$$\min_{\omega_j} \sum_j \mathbf{s}_{ij} \cdot \omega_j^{-\eta}, \quad s.t. \sum_j \omega_j = \varpi,\ \omega_j \geq 0. \tag{5.10}$$

Finally, Eq. (5.10) has a closed-form solution

$$\omega_i = (\frac{\alpha_i^{\frac{1}{1+\eta}}}{\sum_{i=1}^n \alpha_i^{\frac{1}{1+\eta}}})\varpi,$$

where $\alpha_i = \sum_j |\mathbf{s}_{ij}^\tau| \geq 0$ and $\varpi \geq 0$, and the solution will satisfy the non-negative constraints automatically.

Finally, we develop a supervised pipeline of TaDsNet by taking advantage of RNN for Eq. (5.6). Comparing with unsupervised TaDsNet, the loss of supervised TaDsNet is with an additive Mean Square Error (MSE) term to incorporate clinical information. Besides, we add a pooling layer to finalize the features for each subject before the following MSE loss,

$$\arg\min \frac{1}{n} \sum_{i=1}^n ||\mathbf{y}_i - (\mathbf{S}^T\mathbf{S} + \beta\mathbf{I})^{-1}\mathbf{S}^T||_2^2 \tag{5.11}$$

where β is a non-negative sparse coefficient and $\mathbf{y}_i \in \mathbb{R}$ is the ground truth for each response, here is the future time point clinical score. Once TaDsNet converges, the predicted score of $\mathbf{Y}^{t+1}$ with the given multiple time points data $\mathbf{X}^1, \cdots, \mathbf{X}^t$ could be obtained by first passing the input sequence as in Fig. 5.1 into RNN and then solving the above problem Eq. (5.6) with Eq. (5.11) loss.

Table 5.1: The Comparison Results of Predicting 36-month (M36) MMSE and ADAS-cog Scores. (C: Correlation Coefficient and R: Root Mean Square Error)

Methods	MMSE (C)	M36 (R)	ADAS-cog (C)	M36 (R)
TaDsNet	**0.73±0.02**	**2.60±0.79**	**0.75±0.05**	**6.63±1.57**
TaDsNet-L	0.72±0.02	2.66±0.84	0.72±0.02	6.73±1.54
LISTA-L	0.71±0.04	2.83±1.03	0.69±0.04	7.01±0.73
MTSC-L	0.72±0.03	2.78±1.22	0.73±0.04	6.76±1.77
ISTA-L	0.69±0.04	3.25±0.82	0.71±0.05	7.23±0.61

5.3 Experiment

5.3.1 Data and Experimental Settings

We study the performance of TaDsNet on ADNI-1 cohort ($N = 3393$), which consists of five time points structural MR images and responses are the MMSE and ADAS-Cog scores, coming from baseline, 12-, 18-, 24- and 36-months visits. The sample sizes corresponding to five time points are 837, 733, 728, 641 and 454. We use imaging data from the baseline to 24-months to predict 36-months clinical scores.

Image preprocessing: We use hippocampal surface multivariate morphometry statistics (MMS) (Wang *et al.*, 2011b) as learning features, consisting of surface multivariate tensor-based morphometry, which is computed from the conformal grid and describe surface deformation on a local surface region, and radial distance, which measures the surface deformation along the surface normal directions. We use FIRST[1] to segment hippocampi from MR images and follow the same protocol as Shi *et al.* (2013) to extract vertex-wise hippocampal morphometry features, consisting of 4×1 vectors on each vertex of 30000 vertices on every pair of hippocampal surfaces. We select 2000 overlapping surface patches on each pair of hippocampi (1000 on each

[1] `https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FIRST`

side) with patch size of 10×10. After preprocessing the data, we have 1,674,000, 1,466,000, 1,456,000, 1,282,000 surface patches for each input time point matrix $\mathbf{X}^t$ with $p = 400$, respectively.

Comparison Methods: We compare the proposed model with four other methods. TaDsNet is our proposed supervised pipeline; *TaDsNet-L* is unsupervised TaDsNet followed by LASSO; *LISTA-L* is LISTA embeded in a simple RNN, followed by LASSO; *MTSC-L* is the multi-task sparse coding followed by LASSO; *ISTA-L* is the single-task sparse coding followed by LASSO.

Hyperparameters: All experiments are trained on a single Nvidia TITAN X GPU and the same optimization solver is adopted for fair comparisons. TaDsNet takes 200s/epoch while ISTA-L takes 150s/epoch. Therefore, the overall speed is quite affordable for practical use. We set the coefficient parameter $\beta = 0.1$ in Eq. (5.11) based on the grid-search results. We discuss Ω_0 and η selection in Sec. 5.3.2. Besides, we randomly select p samples from matrices $\mathbf{X}^t$ to construct initial dictionaries $\mathbf{D}^t$ for initializing the dictionaries and we set all the sparse codes $\mathbf{S}^t$ to be zero in the beginning and $\kappa = 10$ epochs. We normalize all selected surface patches into $[0, 1]$ and set the size of the dictionary as 1000. For LISTA and RNN, we use the implementation from Zhou *et al.* (2018), with batch size of 512, 10 epoches, initial learning rate (lr) of 0.9 (lr decay 0.95) with momentum factor of 0.9.

5.3.2 Parameter Selection in TaDsNet

We study two affecting parameters – Ω_0 which controls the initial regularization of the sparse codes and η the shrinking factor – based on the performance of our approach in this subsection. We use the F-scores to measure the performance of the variation of the parameters.

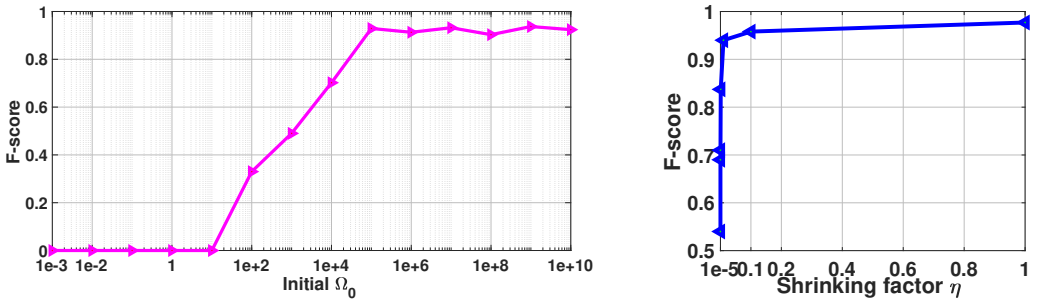


Figure 5.2: The Performance Changes of Different Parameter Selections in TaDsNet.

First, we examine choosing different Ω_0 values for the performance changes from 10^{-3} to 10^{10}. The result is shown on the left side of the Fig. 5.2. We can observe that the performance gradually increases then fails to the fluctuation and the small initial sparsity (less than 10^1) could quickly stop the algorithm at a local minimum. Therefore, we choose the best performance of $\Omega_0 = 10^7$ in all experimental settings.

Second, we test the shrinking factor η from 10^{-5} to 1 (right figure in Fig. 5.2). We observe that the performance becomes stable after sharply increasing. However, the shrinking stages of the system are stable from 0.1 to 1, which means sufficient of the η choice. We thus select $\eta = 0.6$ to balance the efficiency and the quality of the shrinking procedure.

5.3.3 Prediction Results

We evaluate the performance of our proposed pipeline with the comparison methods mentioned above. In the supervised TaDsNet setting, we utilize the MSE results to guide learning sparse codes while RNN is used as a feature extractor for unsupervised TaDsNet. We also compare our method with LISTA, which is a simple RNN without the adaptive-dynamic regularization. In addition, we compare our method with non-RNN embedded SC methods ISTA and MTSC. We report the comparison results for predicting 36-month MMSE and ADAS-cog scores in Table. 5.1. For a fair comparison, we use cross-validation to select sparse parameters from 10^{-3} to 10^3 for

all comparison methods and LASSO. Table. 5.1 shows that TaDsNet and TaDsNet-L outperforms all the baselines (ISTA-L and MTSC-L) by a large margin on both MMSE and ADAS-cog results and this verifies the advantages of TaDsNet. We calculate the non-zero elements of the sparse codes before (20/1000) and after (976/1000) the pooling layer (Fig. 1) of TaDsNet on one single patch. The results demonstrate the supervised loss keeps the sparsity of the proposed approach. Furthermore, TaDsNet-L has better reconstruction power than a simple RNN based optimization method (LISTA-L) due to the adaptive and dynamic learning power. We can also notice that supervised setting of TaDsNet can help improve the results of unsupervised pipeline (TaDsNet-L). It may provide us the insights that the proposed algorithm has a great potential for AD diagnosis and prognosis.

5.4 Summary

In this chapter, I introduce a novel supervised temporal RNN based SC model TaDsNet for modeling AD progression, which adaptively updates the sparse codes and dynamically learns the dictionary atoms. The empirical results on ADNI show the superiority of our model. In our ongoing work, we integrate LSTM (Zhou *et al.*, 2018) with our model to further improve the convergence rate of TaDsNet.

Chapter 6

DEEP MULTI-ORDER PRESERVING WEIGHT CONSOLIDATION

6.1 Introduction

Traditional machine learning algorithms have been widely applied on AD progression modeling. Stonnington *et al.* (2010) predict clinical scores by using relevance vector regression model and Sukkar *et al.* (2012) use hidden Markov chains to model AD progression, but these works predict the target clinical scores at an isolated single time point. Therefore, many researchers develop joint analysis schemes on multiple time points data to improve the performance of the single-task. Zhou *et al.* (2013) use a convex fused sparse group lasso model to predict AD clinical scores at different time points and Zhang *et al.* (2017c) use a multi-task dictionary learning framework to predict clinical scores of multiple time points, but they both ignore the time-order information along multiple tasks. Recently, Wang *et al.* (2018) model the AD progression via Recurrent Neural Network (RNN). Although the RNN models the progression in time order, it does not take into account that the longitudinal image data are obtained in sequences rather than a uniform batch mode (e.g., the MR images are taken in different time points, not in one time only). None of above-mentioned algorithms consider the real-world scenarios. It motivates us to develop a lifelong learning system to mimic how doctors monitor and prognosticate the AD progression.

Recently, deep neural networks have brought breakthroughs in various medical imaging studies, such as object classification (Suk *et al.*, 2014), segmentation (Huo *et al.*, 2016), medical image diagnosis (Zhang *et al.*, 2017e), etc. However, the algorithms still can be further improved, because in real-world disease diagnosis applica-

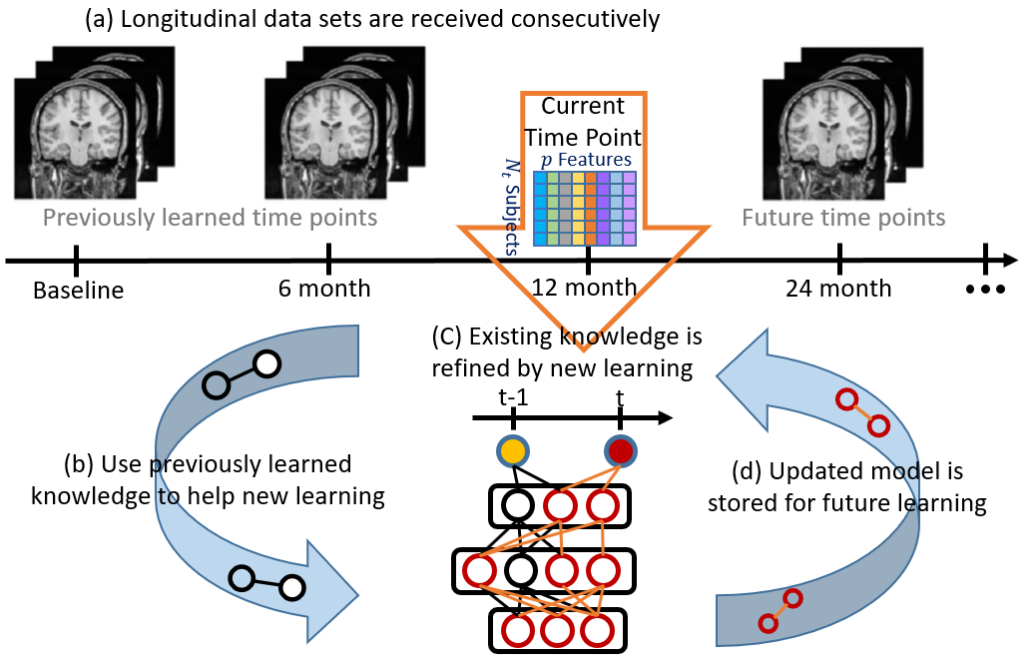


Figure 6.1: Overview of Proposed Lifelong Longitudinal Feature Learning Framework.

tions, different batches of data arrive periodically (e.g., monthly, seasonally, or yearly) with the data distribution changing over time rather than all data coming together. This presents an opportunity for lifelong learning, whose primary goal is to learn consecutive tasks without forgetting the knowledge learned in the past (e.g. with less longitudinal data) and leverage the previous knowledge to build a lifelong learning machine to achieve artificial general intelligence. One simple way is to fine-tune the model for every new data set; however, the retrained representations may adversely affect the old tasks, causing them to drift from their optimal solution. This can cause "catastrophic forgetting", a phenomenon where training a model with new tasks interferes the previously learned old knowledge, leading to a performance degradation or even overwriting of the old knowledge by the new ones.

To overcome the above "catastrophic forgetting" problem, many approaches have been proposed (Kirkpatrick *et al.*, 2017; Li and Hoiem, 2017; Lopez-Paz *et al.*, 2017). Kirkpatrick *et al.* (2017) propose using a regularization term to prevent the new

weights from deviating too much from the previously learned weights, based on their significance to old tasks. Learning without forgetting (LwF) (Li and Hoiem, 2017) leverages distillation regularization on the new tasks — the soft labels of previously learned tasks are enforced to be similar to the network with the current task by using knowledge distillation (Hinton *et al.*, 2015). Gradient of Episodic Memory (GEM) (Lopez-Paz *et al.*, 2017) uses episodic memory, where the previously learned task samples are stored to effectively recall the experience in the past, and learns the subset of correlations to a set of tasks without using task descriptors. However, none of the existing lifelong learning methods considers the discrimination weight subset by incorporating inherent correlations between old tasks and new tasks. We therefore propose a novel lifelong longitudinal feature learning algorithm to learn the longitudinal data in sequence, and leverage the inner and inter relationship between learned model and incoming data to achieve a general longitudinal image data analysis.

Although the lifelong learning may learn the longitudinal data in time sequence manner, it is important to respect the valuable temporal information from the longitudinal data coming in time order (e.g. patient's 3-month MR image comes in front of 12-month MR image). We therefore design a time-order preserving term which may ensure features at a certain time point be temporally ahead of those of succeeding time points. In this chapter, we develop a multi-task based lifelong learning framework termed Multi-order Preserving Weight Consolidation (dMopWC), to continually learn on time-order sequential longitudinal data without losing statistical power on less longitudinal data and ensure that the temporal information is respected in the lifelong learning solution. Fig. 6.1 shows the overview of proposed lifelong learning framework.

The key contributions of this work can be summarized in threefold. Firstly, we formulate the disease progression in a lifelong learning manner which respects the lon-

gitudinal data sets coming in sequence. To the best of our knowledge, it is the first learning model which models disease progression in continually sequential manner and continually predict future cognitive decline with brain imaging analysis. Secondly, to overcome "catastrophic forgetting" for the old learned time points' information, we propose a novel Multi-order Preserving Weight Consolidation (dMopWC) — it considers the discriminative weight subset by incorporating inherent correlations between old and new time points' information and learns the task-specific patient's information for the new time point. Thirdly, unlike previous lifelong learning algorithms, we take the time order knowledge of longitudinal data into consideration and formulate a time-order preserving term to ensure the temporal information is respected in our solution. Our extensive experimental results on Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset show the proposed dMopWC achieves higher correlation coefficients and lower estimation error, as well as more linear and reasonable prediction scores when comparing with other single-task, multi-task and linear/non-linear algorithms.

6.2 Method

Fig. 6.1 is an overview of our lifelong longitudinal feature learning framework. It has three key characteristics: learn longitudinal data consecutively, store previously learned time points' knowledge and update the stored model with new time point data. The pipeline source code is publicly available at `https://github.com/zj00377/DMopWC`.

6.2.1 Problem Definition and Overview

We define the lifelong learning problem as follows — there will be an unknown number of MR images belonging to different tasks (time points) with unknown dis-

tributions, arriving in sequence. The task can be a single task or multiple different tasks (e.g., patients' images from a single time point or multiple time points). Our goal is to learn a deep model in such a lifelong learning scenario without "catastrophic forgetting". At testing time, the task at time point t will be given and we aim to test the future clinical scores from time point $t+1$. Given a sequence of T tasks, task at time point $t = 1, 2, \cdots, T$ with N_t images comes with dataset $\mathbf{D}_t = \{\mathbf{x}_i^t, y_i^t\}_{i=1}^{N_t}$. Specifically, for task t, y_i^t is the ground truth of the clinical scores for the i-th subject $\mathbf{x}_i^t \in \mathbb{R}^p$ at time point t. We denote the training data matrix by $\mathbf{X}^t$ for $\mathbf{D}_t$, i.e., $\mathbf{X}^t = (\mathbf{x}_1^t, \cdots, \mathbf{x}_{N_t}^t)$. When the dataset of time point t comes, all the previous training time points' datasets $\mathbf{D}_1, \cdots, \mathbf{D}_{t-1}$ are not available any more, but the deep model parameter with l layers $\theta^{t-1} = \{\theta_l^{t-1}\}_{l=1}^L$ can be accessed. The lifelong learning problem at time point t when given data $\mathbf{D}_t$ can be defined as solving the following problem:

$$\min_{\theta^t} \mathcal{L}(\theta^t | \theta^{t-1}, \mathbf{D}_t) + \lambda \Omega(\theta^t), t = 1, \cdots, T \tag{6.1}$$

where $\mathcal{L}$ is the loss function of solving θ^t, θ^t is the model parameters for time point t. $\Omega(\cdot)$ can include one or more non-smooth sparsity-inducing norms and λ is a non-negative parameter. Note that the number of the upcoming time point data sets can be finite or infinite — for simplification, we consider the finite scenario here.

Kirkpatrick *et al.* (2017) proposed Elastic Weight Consolidation (EWC) to solve the above problem (6.1) that consists of a quadratic penalty on the difference between the parameter θ^t and θ^{t-1} to slow down the "catastrophic forgetting" for previously learned time point information. The posterior distribution $p(\theta^t | \mathbf{D}_t)$ is used to describe the problem by the Bayes' rule,

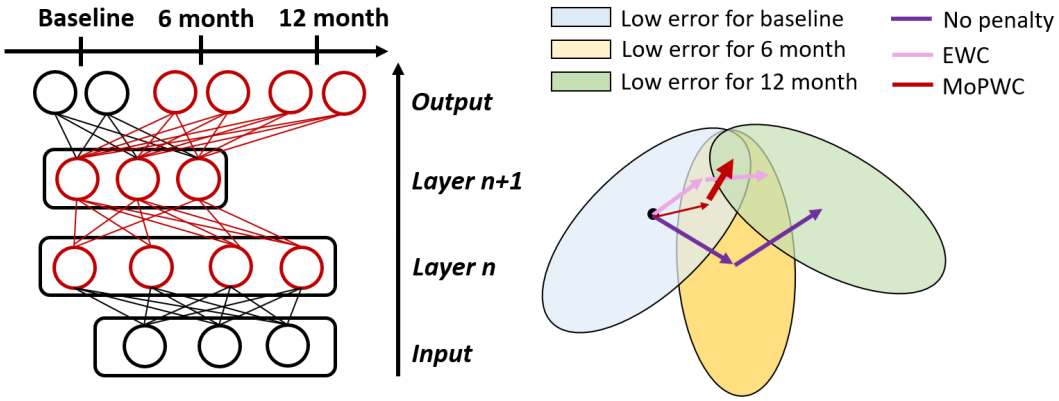


Figure 6.2: Graphical Illustration of the Proposed Multi-Order Preserving Weight Consolidation (dMopWC). dMopWC First Learns a Model on Baseline Data (Blue), Then Updates It After Observing 6-month Data (Yellow) and Finally Updates the Updated Model after Learning 12-month Data (Green). The Thicker Red Arrow Denotes Larger Time-order Penalty on Later Time Point. dMopWC can Keep Most Previously Learned Knowledge Comparing with EWC and Fine-tuning.

$$\log p(\theta^t|\mathbf{D}_t) = \log p(\mathbf{D}_t|\theta^t) + \log p(\theta^t|\mathbf{D}_{t-1}) - \log p(\mathbf{D}_t), \tag{6.2}$$

where the posterior probability $\log p(\theta^t|\mathbf{D}_{t-1})$ embeds all the information from task $t-1$. However, the problem (6.2) is intractable so that EWC approximates it as a Gaussian distribution with mean of parameter $\bar{\theta}^{t-1}$ and a diagonal matrix I of the Fisher Information matrix $\mathbb{F}$. The Fisher information matrix $\mathbb{F}$ is computed by

$$\mathbb{F}_i^t = I(\theta^t)_{ii} = E_x[(\frac{\partial}{\partial \theta_i^t} \log p(\mathbf{D}_t|\theta^t))^2|\theta^t]. \tag{6.3}$$

Therefore, the problem of EWC at time point t can be rewritten as follows:

$$\min_{\theta^t} \quad \mathcal{L}_t(\theta^t) + \frac{\lambda_1}{2} \sum_i \mathbb{F}_i^{t-1}(\theta_i^t - \bar{\theta}_i^{t-1})^2, \tag{6.4}$$

where $\mathcal{L}_t$ is the loss function for time point t, λ_1 denotes how important the time point $t-1$ data is compared to time point t data and i labels each weight of the parameter θ.

6.2.2 *Multi-order Preserving Weight Consolidation*

The main problem of EWC is that EWC only enforces time point t data close to time point $t-1$ data. This will ignore the patient's inherent correlations within time point t and the same patient's information between time point $t-1$ and time point t and such relationship might potentially help improve the statistical power and overcome "catastrophic forgetting" on the previously learned time points' information. Learning multiple related time points' data jointly can improve performance relative to learning each time point data separately, when the two time points' data are related — this idea has been incorporated into Multi-Task Learning (MTL) (Evgeniou and Pontil, 2004). It has been commonly used to obtain better generalization performance than learning each task individually. One appealing property of the $l_{2,1}$-norm regularization is that it shares similar parameter sparsity patterns among multiple different tasks. Therefore, the MTL via the $l_{2,1}$-regularization can be incorporated into Eq. 6.4 and the objective function of multi-task based elastic weight consolidation can be written into Eq. 6.5 to improve the ability of overcoming "catastrophic forgetting" from multiple time points and enforce the sparsity over features for multiple time points simultaneously,

$$\min_{\theta^t} \mathcal{L}_t(\theta^t) + \frac{\lambda_1}{2} \sum_i \mathbb{F}_i^{t-1}(\theta_i^t - \bar{\theta}_i^{t-1})^2 + \lambda_2 \sum_i ||\theta_i^t||_{2,1}, \tag{6.5}$$

where λ_2 is the non-negative regularization parameter and $||\theta_i^t||_{2,1} = \sum_j ||\theta_{i,j}^t||_2$ is the $l_{2,1}$-norm regularization to learn the related representations and j presents j-th subject/row. Here, we employ the multi-task learning with $l_{2,1}$-norm to capture the common subset of relevant parameters from time point t subjects and it enforces the important features to have non-zero weights cross all subjects. However, some of the important features might be outliers of the feature space and need to be paid special attention.

Specifically, we further consider some important parameters which have better representation power to a subset of the time point data set. The MTL with sparsity-inducing norm (Gong *et al.*, 2012) has been widely studied to select such discriminative parameter subset by incorporating inherent correlations among multiple subjects. It has been shown that l_1 sparse norm (Liu *et al.*, 2014) can identify informative longitudinal phenotypic biomarkers that are related to pathological changes of AD in brain image analysis. To this end, the l_1 sparse norm is imposed to learn the discriminative new task-specific parameters while learning task relatedness among multiple time points' tasks. Therefore, the objective function for time point t becomes:

$$\min_{\theta^t} \quad \mathcal{L}_t(\theta^t) + \frac{\lambda_1}{2}\sum_i \mathbb{F}_i^{t-1}(\theta_i^t - \bar{\theta}_i^{t-1})^2 + \lambda_2 \sum_i ||\theta_i^t||_{2,1} + \lambda_3||\theta^t||_1, \tag{6.6}$$

where λ_3 is the non-negative regularization parameter. Eq. 6.6 studies the *discriminative weights subset* with inherent correlations among *multiple time point tasks* while keeping previously learned time points' knowledge via weight consolidation.

Although some existing lifelong learning models may be applied to study brain longitudinal images, how to utilize the time ordering imaging information remains an open problem. Here, we introduce a novel time-order preserving criteria to enrich lifelong learning models. The goal of the longitudinal order preserving is to prevent the time point t information θ^t from being temporally in front of the time point $t-1$ information of θ^{t-1}. For instance, for longitudinal data, we know that 3-month visit is behind baseline visit and 12-month visit is behind 3-month visit and baseline visit (See Fig. 6.1). In other words, the lifelong learning model observes the same temporal order as the input longitudinal time series. Thus, we introduce the expression,

$$w^t||\theta^t - \theta^{t-1}||_2^2, \tag{6.7}$$

where w^t represents the temporal order weight function for time point t. Therefore, $w^{t-1}\theta^{t-1} < w^t\theta^t$ represents the approximated temporal order of the time point t. In

Algorithm 8: Multi-order Preserving Weight Consolidation (dMopWC)

Input : Longitudinal dataset $\mathbf{D}_1, \cdots, \mathbf{D}_T$; $\lambda_1, \lambda_2, \lambda_3, \lambda_4$

Output: θ^T

```
begin
    for t = 1 → T do
        if t = 1 then
            Train an initial network with weights θ^1 by using Eq. 6.1 on D_1.
            Computer Fisher information matrix F_i^1 by using Eq. 6.3 on D_1.
        else
            According to F_i^{t-1} and θ^{t-1}, optimize θ^t by using Eq. 6.9 and D_t.
            Computer Fisher information matrix F_i^t by using Eq. 6.3 on D_t.
```

this work, we choose a simple element-wise linear form of the weight function w to reflect the longitudinal time ordering information as follows:

$$w = [\frac{1}{T}, \frac{2}{T}, \cdots, \frac{t}{T}, \cdots, \frac{T-1}{T}, 1]. \tag{6.8}$$

Therefore, the final objective function of the proposed Multi-order Preserving Weight Consolidation (dMopWC) will become

$$\min_{\theta^t} \mathcal{L}_t(\theta^t) + \frac{\lambda_1}{2} \sum_i \mathbb{F}_i^{t-1} (\theta_i^t - \bar{\theta}_i^{t-1})^2 + \lambda_2 \sum_i ||\theta_i^t||_{2,1} + \lambda_3 ||\theta^t||_1 + \lambda_4 w^t ||\theta^t - \theta^{t-1}||_2^2, \tag{6.9}$$

where λ_4 is an non-negative parameter. Fig. 6.2 shows the geometric illustration of dMopWC, it shows that our method can learn the most common sub-area (three colors' overlapping area) among three time points' data and preserve time-order in sequence comparing with EWC (two colors' overlapping area). The left figure in Fig. 6.2 illustrates that dMopWC has the same model size across multiple time points learning.

6.2.3 Optimization

Eq. 6.9 is a convex non-smooth objective function, we introduce the details of optimizing each of its terms in this subsection. For the first term in Eq. 6.9, the standard choice for $\mathcal{L}(\cdot)$ is the mean squared error for regression problem at time point t,

$$\mathcal{L}_t^{MSE} = \frac{\sum_{i=1}^{N_t} ||y_i^t - \hat{y}_i^t||_2^2}{N_t}, \tag{6.10}$$

where y_i^t is the ground truth and $\hat{y}_i^t$ is the prediction. N_t denotes the number of subjects.

The third and fourth terms in Eq. 6.9 are convex non-smooth term used to regularize the model. For the l_1-norm, one popular variation is to approximate the l_1-norm by a convex term,

$$\Omega_1 = ||\theta^t||_1 = \sum_l \sqrt{\theta_l^{t2} + \beta}, \tag{6.11}$$

where l denotes each layer of θ^t and β is a sufficiently small scalar factor to obtain a smooth problem. For the group sparse $l_{2,1}$ regularization term, it can be written as (Scardapane *et al.*, 2017),

$$\Omega_{2,1} = ||\theta_i||_{2,1} = \sum_j \sqrt{|\theta_{i,j}|}\ ||\theta_{i,j}||_2, \tag{6.12}$$

where $|\theta_{i,j}|$ is the dimension of the $\theta_{i,j}$ and it ensures that each group gets weighted uniformly. However, the gradient when $||\theta_{i,j}|| = 0$ is not defined in $\Omega_{2,1}$, the formulation might still be sub-optimal. Thus, we give the sub-grient of Eq. 6.12 as follows,

$$\frac{\partial \Omega_{2,1}}{\partial \theta_{i,j}} = \begin{cases} \sqrt{|\theta_{i,j}|}\ \frac{\theta_{i,j}}{||\theta_{i,j}||_2} & if ||\theta_{i,j}|| \neq 0, \\ \sqrt{|\theta_{i,j}|}\ g:\ ||g||_2 \leq 1 & otherwise. \end{cases} \tag{6.13}$$

$\Omega_{2,1}$ and Ω_1 terms are so-called "Sparse Group Lasso" penalty (Scardapane *et al.*, 2017; Zhou *et al.*, 2013), the optimal results can be achieved by considering a single regularization factor for both terms.

Recall the second and fifth terms in Eq. 6.9, the EWC and the time-order preserving regularization are both weighted least square function. It is obviously that the least square function is convex and continuously differentiable term. We summarize the steps of our dMopWC algorithm in Algorithm 8. We first learn a basic model on the baseline dataset, and learn the sequential longitudinal data sets by solving Eq. 6.9.

6.3 Experiments

6.3.1 Data and Experimental Settings

Datasets. We evaluate our dMopWC algorithm on the entire ADNI-1 cohort (Jack Jr *et al.*, 2008) for lifelong learning. We study seven time points structural MR Images and responses are MMSE and ADAS-Cog scores, coming from seven different time points: baseline, M06, M12, M18, M24, M36 and M48. The sample sizes corresponding to seven time points are 837, 733, 728, 326, 641, 454 and 251. Specifically, we remove 25 subjects without MMSE and ADAS-cog from baseline data and we use 812 subjects instead. The hippocampal surface multivariate morphometry statistics (MMS) (Wang *et al.*, 2011b) are utilized as learning features, consist of surface multivariate tensor-based morphometry, which is computed from the conformal grid and describe surface deformation on a local surface region, and radial distance, which measures the surface deformation along the surface normal direction. We use FIRST[1] to segment hippocampi from MR images and follow the same protocol as Shi *et al.* (2013) and extract vertex-wise hippocampal morphometry features, consisting of 4×1 vectors on each vertex of 30000 vertices on every pair of hippocampal surfaces. As a result, each subject $\mathbf{x}_i^t$ has $p = 120,000$ features in total. In the prediction, we use

[1] `https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FIRST`

the current time point data to predict future clinical score, e.g., we study baseline MR images and predict 12-month MMSE/ADAS-cog.

Network settings. We use a two-layer fully-connected neural network of 100-100 units with ReLU activations as our initial network. All comparison algorithms are trained on a single Nvidia TITAN X GPU. All models and algorithms are implemented using Tensorflow[2] library. We will release our code on our website for comparison purpose upon the acceptation of the paper.

Hyperparameter settings. All hyper-parameters in dMopWC are optimized using grid-search and the best results for each model are reported. The SGD optimizer is used with a learning rate of 0.001 and we set batch size of 256 with 1400 iterations, $\lambda_1 = 15$, $\lambda_2 = 0.0001$, $\lambda_3 = 0.15$ and $\lambda_4 = 0.5$ on MMSE and $\lambda_1 = 13$, $\lambda_2 = 0.015$, $\lambda_3 = 0.00001$ and $\lambda_4 = 0.1$ on ADAS-cog. We use 200 subjects to compute $\mathbb{F}_i^t$.

Evaluation methods. In order to evaluate the proposed model, we randomly split the data into training and testing sets using a 9:1 ratio and repeat this procedure 20 times to avoid data bias. We report the mean and standard deviation of these 20 different splits. Lastly, we evaluate the overall regression performance using weighted correlation coefficient (wR) and root Mean Square Error (rMSE) for task-specific regression performance measures. The two measures are defined as $wR(\mathbf{Y}, \hat{\mathbf{Y}}) = \sum_{t=1}^{T} Corr(\mathbf{Y}_t, \hat{\mathbf{Y}}_t)$
$N_t / \sum_{t=1}^{T} N_t$ and $rMSE(\mathbf{Y}_t, \hat{\mathbf{Y}}_t) = \sqrt{||\mathbf{Y}_t - \hat{\mathbf{Y}}_t||_2^2 / N_t}$, where $Corr$ is the correlation coefficient between two vectors and N_t is the number of subjects of task t. $\mathbf{Y}_t$ and $\hat{\mathbf{Y}}_t$ are the ground truth of targets and the corresponding prediction at time point t, respectively. The smaller rMSE and the larger wR represent the better results.

Comparison methods. We compare our algorithm with three groups of methods: single-task regression methods: 1) LASSO regression (Tibshirani, 1996a). 2)

[2] `https://www.tensorflow.org/`

Ridge regression (Hoerl and Kennard, 1970a); multi-task regression methods: 1) L21: the multi-task method named $L_{2,1}$ norm regularization with least square loss (Liu *et al.*, 2014). 2) cFSGL: the multi-task method called convex fused sparse group Lasso (Zhou *et al.*, 2013), 3) MSMT: the multi-soure multi-target dictionary learning (Zhang *et al.*, 2017c); deep learning methods: 1) SN: a single deep network trained across all time points data. 2) EWC (Kirkpatrick *et al.*, 2017): a deep network trained with elastic weight consolidation. 3) dMopWC: the proposed algorithm. For linear regression methods, cross validation is used to select the model parameters in the training data and the same baseline and 6 month images features are used to predict future clinical scores. We select patches as inputs for MSMT as (Zhang *et al.*, 2017c). For deep learning methods, we use the same setting of the initial model and the sequential data is used to predict its 12-month later clinical scores.

Table 6.1: The Prediction Results of MMSE on ADNI-I Dataset.

Methods	wR	M12	M18	M24	M36	M48
Lasso	0.40±0.09	4.04±0.77	3.46±0.97	5.53±0.86	4.39±0.74	4.73±1.49
Ridge	0.41±0.07	4.26±0.56	3.56±0.93	5.05±0.54	4.21±0.47	3.62±0.91
L21	0.57±0.01	3.32±0.63	4.75±0.75	4.64±0.88	4.08±1.01	3.11±1.05
cFSGL	0.72±0.03	2.67±0.32	3.40±0.99	3.64±0.71	2.98±0.81	2.60±1.13
MSMT	0.73±0.02	2.61±0.55	3.37±1.01	3.66±0.78	2.73±1.09	2.52±1.20
SN	0.72±0.05	**2.54±0.23**	2.05±0.15	2.09±0.26	2.32±0.23	2.21±0.13
EWC	0.71±0.09	2.60±0.33	2.41±0.28	1.98±0.41	2.93±0.22	2.65±0.39
dMopWC	**0.76±0.04**	2.58±0.24	**1.75±0.30**	**1.90±0.26**	**2.30±0.19**	**2.03±0.12**

Table 6.2: The Prediction Results of ADAS-cog on ADNI-I Dataset.

Methods	wR	M12	M18	M24	M36	M48
Lasso	0.49±0.05	6.81±1.03	6.87±0.74	7.62±0.87	8.08±1.39	6.55±1.34
Ridge	0.46±0.07	7.68±0.96	6.89±1.69	7.84±1.54	8.59±0.62	6.64±1.58
L21	0.53±0.07	6.40±0.51	6.95±0.88	8.07±0.67	8.00±1.04	5.92±0.60
cFSGL	0.73±0.04	5.32±0.97	5.27±0.66	6.79±1.00	6.34±1.11	5.61±0.82
MSMT	0.77±0.02	5.18±0.88	**4.64±1.12**	6.76±1.35	6.78±1.54	**5.27±1.76**
SN	0.72±0.04	5.18±0.59	5.31±0.67	5.68±0.65	5.87±0.69	5.95±0.23
EWC	0.75±0.06	5.22±0.47	5.24±0.81	5.55±0.12	5.78±0.18	5.94±0.43
dMopWC	**0.78±0.03**	**5.18±0.26**	5.19±0.25	**5.40±0.21**	**5.66±0.23**	5.71±0.10

6.3.2 Experimental Results

Performance Comparisons. We report the results of dMopWC and other methods on the prediction model of MMSE on ADNI-I dataset in Table 6.1. The proposed approach dMopWC outperforms single-task regression methods, in terms of both rMSE and correlation coefficient on four different time points. The results of Lasso and Ridge are very close while multi-task regression methods are superior to them. For multi-task regression models, we observe that dictionary learning based algorithm obtains lower rMSE and higher correlation results than other multi-task regression methods. We also notice that the deep learning algorithms strongly improve the prediction results over linear regression algorithms. The proposed dMopWC has better performance than SN because the retraining of SN does not consider the knowledge of previous time points. SN has better performance than EWC and dMopWC on M12 due to random initialization of the weights of deep neural networks on baseline data, but M12 values of three methods are really close comparing with other time points' results. However, EWC has worse performance than SN on most time points while dMopWC significantly improve EWC because it studies the time-order information along with common weight subset and discriminative new time point features while keeping the old time points' knowledge. Besides, dMopWC enhances the results of cFSGL because of the knowledge accumulation of the previous time points.

We follow the same experimental settings in the MMSE study and explore the prediction model by ADAS-cog scores. The performance results are reported in Table 6.2. We can observe that the best performance of predicting scores of ADAS-Cog is achieved by dMopWC in three time points. MSMT has smallest rMSE on M18 and M48 because of the fluctuation of scores when the available amount of data becomes less. However, after dMopWC dealing with temporary sequence information, the re-

sults are more linear, reasonable and accurate on all time points. We also find out that the proposed dMopWC has much more improvement on M24 and M36 than MMSE prediction. Since we keep the previous time points' knowledge, the later time points do not have bias comparing with linear regression algorithms. Overall, dMopWC has more reasonable results on both clinical scores because modeling disease progression via lifelong learning can accumulate the early stage knowledge for later time points.

Table 6.3: Ablation Study Results of MMSE on ADNI-I Dataset.

Methods	wR	M12	M18	M24	M36	M48
EWC	0.71±0.09	2.60±0.33	2.41±0.28	1.98±0.41	2.93±0.22	2.65±0.39
EWC+l_1	0.72±0.08	2.59±0.35	2.14±0.33	2.16±0.47	2.73±0.54	2.20±0.19
EWC+l_{21}	0.73±0.04	2.62±0.28	2.04±0.21	2.09±0.48	2.35±0.25	2.31±0.21
dMopWC w/o oP	**0.74±0.02**	**2.58±0.17**	**1.98±0.11**	**1.96±0.23**	**2.32±0.20**	**2.16±0.17**

Table 6.4: Ablation Study Results of ADAS-cog on ADNI-I Dataset.

Methods	wR	M12	M18	M24	M36	M48
EWC	0.75±0.06	5.22±0.47	5.24±0.81	5.55±0.12	5.78±0.18	5.94±0.43
EWC+l_1	0.75±0.08	5.24±0.25	5.34±0.29	5.38±0.42	5.73±0.45	5.93±0.26
EWC+l_{21}	0.76±0.04	**5.20±0.39**	5.37±0.32	5.48±0.39	5.68±0.40	5.76±0.23
dMopWC w/o oP	**0.77±0.02**	5.23±0.32	**5.21±0.18**	**5.42±0.24**	**5.66±0.21**	**5.73±0.16**

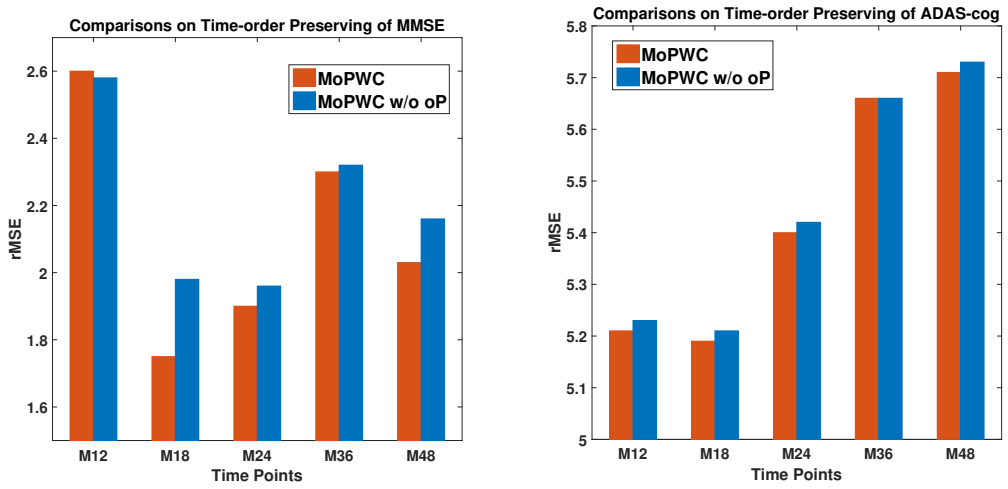


Figure 6.3: Comparisons on Time-order Preserving Term of rMSE Performance on ADNI-I Dataset.

Ablation study of different terms in dMopWC. We study how the different components used in dMopWC affect the final performance of lifelong learning. We report the MMSE and ADAS-cog prediction results on ADNI-I dataset of different strategies *EWC*, *EWC with* l_1*-norm only*, *EWC with* l_2*-norm only* and *dMopWC without order-preserving term* in Table 6.3 and Table 6.4. It shows that $l_{2,1}$-norm has a stronger effect of the performance than l_1-norm while our method dMopWC without order-preserving term outperforms the single regularized term strategies. The results demonstrate the effectiveness of our method by studying common weights subset with discriminative new time points information.

Effect of time-order perserving term. We also compare the effectiveness of the time-order preserving term against the dMopWC without order-preserving term. Fig. 6.3 shows the comparison results. dMopWC achieves better rMSE performance than dMopWC w/o oP. Fig. 6.3 demonstrates dMopWC further improves the results because we consider the time order smoothness problem in longitudinal dataset, especially dMopWC significantly improves the result of M48. This may be due to the baseline data has less correlation with later time points' data and dMopWC w/o

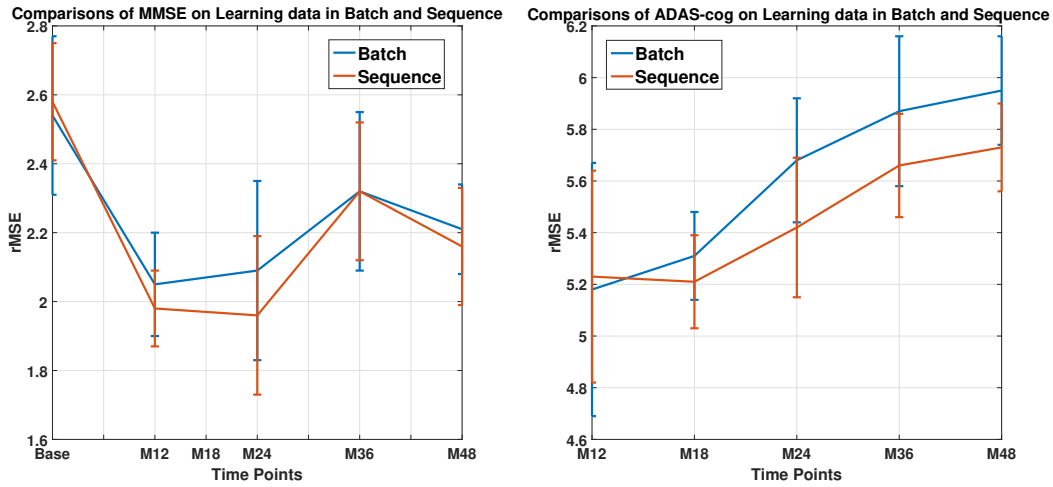


Figure 6.4: Comparisons of rMSE Performance on MMSE and ADAS-cog When Learn Data in Batch and Sequential Mode.

oP assumes each time point has the same correlation for the later time points. Although more validations are warranted, our experimental results show our temporal order-preserving formulation offers a unique perspective on prognosis with longitudinal data.

Comparisons of learning data in batch mode and sequential mode. We study the difference between learning longitudinal data in batch mode and sequential mode in Fig. 6.4, which shows the rMSE values of MMSE and ADS-cog. We can observe that the performance of learning sequential longitudinal data is better than that of learning each time point data in batch. It may partial due to the fact that the model will keep the previous time points' knowledge and learn the new time point information to improve future results when we learn the longitudinal data via lifelong learning. However, learning all images together ignores the relationship of early time points and cannot be fully taken advantage by latest time points to boost the later time points' prediction results.

6.4 Summary

In this chapter, we propose a novel multi-task based lifelong longitudinal feature learning to predict future cognitive decline, termed Multi-order Preserving Weight Consolidation (dMopWC). dMopWC studies inner and inter biomarkers for multiple time points while overcoming "catastrophic forgetting" when learn the sequential tasks. We also consider the time-order preserving problem for longitudinal data. The effectiveness of the proposed algorithm is supported by extensive experimental studies on a relatively large brain imaging cohort - ADNI. They demonstrate that the proposed progression model is more effective than some other state-of-the-art methods. In the next chapter, I extend this algorithm to general computer vision problem and study on four benchmark datasets to further improve the current work.

Chapter 7

REGULARIZE, EXPAND AND COMPRESS: NONEXPANSIVE CONTINUAL LEARNING

7.1 Introduction

In many real-world applications, batches of data arrive periodically (e.g., daily, weekly, or monthly) with the data distribution changing over time. This presents a challenge for continual learning (CL), and is an important topic of study in machine learning. The primary goal of continual learning is to learn consecutive tasks without forgetting the knowledge learned from earlier tasks, and leverage the previous knowledge to obtain better performance or faster convergence on the new tasks. One naive way is to fine-tune the model for every new task; however, such retraining typically degenerates the model performance on both new tasks and the old ones. If the new tasks are greatly different from the old ones, we might not be able to obtain the optimal model for the new tasks. Meanwhile, the retraining may adversely affect the old tasks, causing them to drift from their optimal solution. This is known as "catastrophic forgetting", a phenomenon where training a model with new tasks interferes the previously learned old knowledge, leading to a performance degradation or even overwriting of the old knowledge by the new one.

To overcome above catastrophic forgetting problem, many approaches have been proposed (Kirkpatrick *et al.*, 2017; Li and Hoiem, 2017; Rebuffi *et al.*, 2017; Zhang and Wang, 2019b; Zhang *et al.*, 2020). Kirkpatrick *et al.* (2017) propose using a regularization term to prevent the new weights from deviating too much from the previously learned weights, based on their significance to old tasks. Their method

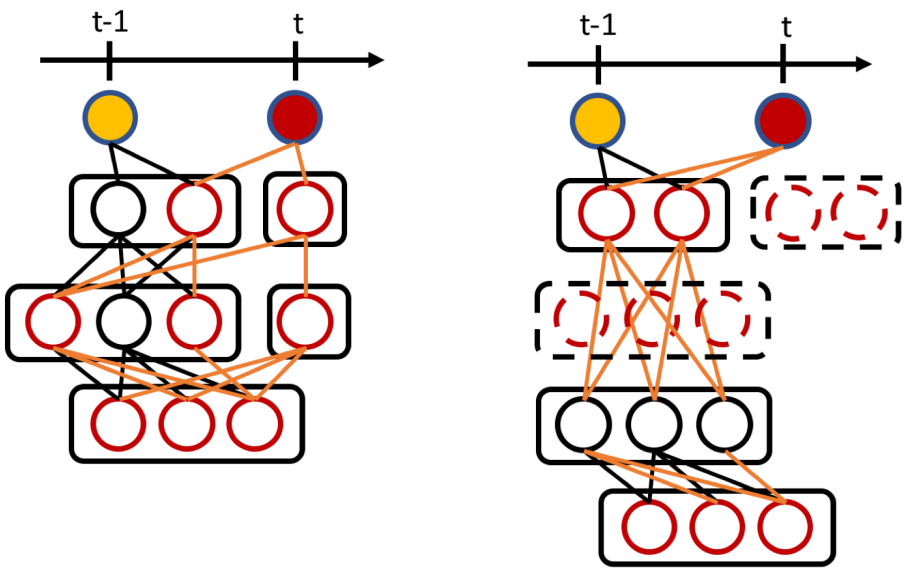


Figure 7.1: (a) The Previous State-Of-The-Art CL Method, DEN (Yoon *et al.*, 2017), Selectively Retrains the Old Network, and Dynamically Expands the Model Capacity. (b) The Proposed Method Expands the Network through Network Transformation based AutoML, and Then Subsequently Compresses the Model Back to Its Original Size.

uses a fixed neural network architecture, which would not scale up when network capacity gets saturated with more and more new tasks to learn. Dynamically expanding the network (Yoon *et al.*, 2017) (DEN) is one way to overcome the problem caused by static architecture — it expands the network capacity whenever it detects that the loss for the new task would not reach a pre-defined threshold. However, DEN involves many hyperparameters and the final performance is highly sensitive to these parameters; it relies on hand-crafted heuristics to explore the tuning space. This search space can be considerably large, and human experts usually find a sub-optimal solution in a time-consuming parameters tuning process. To this end, we aim to automatically expand the network for CL, with better performance and less parameter redundancy than human-designed architectures. To better facilitate automatic knowledge transfer without human expert tuning and model design with optimized model complexity,

we unprecedentedly propose a regularized nonexpansive CL framework while taking learning efficiency into consideration.

AutoML refers to automatically learn a suitable machine learning (ML) model for a given task — Neural Architecture Search (NAS) (Zoph and Le, 2016) is a subfield of AutoML for deep learning, which searches for optimal hyperparameters of designing a network architecture using reinforcement learning (RL). The RL framework has a main controller that observes the generated children networks' performance on the validation set as the reward signal — it then assigns a high probability to the architecture candidate that have high validation accuracy to update the model. If we use this approach directly in the continual learning setting, it would forget old tasks' knowledge, and it would be a wasteful process since each new task network architecture has to be searched from scratch by the controller, ignoring the correlations between previously learned tasks and the new task. We hereby propose a regularized weight consolidation (RWC) approach to obtain an effective classifier by exploiting inherent correlations between old tasks and new task. Furthermore, to narrow down the architecture search space and save time, network transformation (Chen *et al.*, 2015) is utilized to accelerate meta-learning of the new network.

However, if we keep expanding the network for more and more new tasks, the model size will grow drastically to violate piratical efficiency requirements (e.g., low memory footprint, low power usage). Many network-expansion-based continual learning algorithms (Rusu *et al.*, 2016; Yoon *et al.*, 2017) increase the model capability but also decrease the learning efficiency in terms of memory cost and power usage. Therefore, we conduct model compression after completing the learning of each new task — we compress the expanded model to the initial model size (before network expansion), with negligible performance loss on both old and new tasks. Fig 7.1 illustrates the main difference of our approach with network-expansion-based continual

learning algorithms.

In this work, we focus on 1) overcoming catastrophic forgetting for CL and 2) improving the network capacity without decreasing learning efficiency. We propose a new sparse group *regularized* weight consolidation (RWC), to address the first problem. Compared to previous works, e.g. EWC (Kirkpatrick *et al.*, 2017), RWC can identify and retrain discriminative subset of parameters by incorporating inherent correlations among multiple learned new tasks and extract more meaningful features from old tasks, while EWC only considers the previous tasks' Fisher Information. The experimental results show RWC achieves higher average per-task accuracy compared to EWC, especially later tasks. To address the second problem, we aim to automatically *expand* the network for CL with high performance and optimized model complexity without human expert tuning.

We therefore consider the newly expanded layer as a new task-specific layer, where l_1 regularization is adopted to promote sparsity for the new weight so that each neuron only connects with few neurons in the following layer. This will efficiently learn a discriminative representation for the new task while reducing the computation overheads. We then *compress* the expanded model to the same model size as the initial model, with negligible performance loss on both old and new tasks. This is different from previous network-expansion-based CL algorithms, e.g., DEN (Yoon *et al.*, 2017) and PGN (Rusu *et al.*, 2016), which reduce the model efficiency after learning new tasks. As far as we know, this is the first regularization-based nonexpansive AutoML algorithm for CL.

The key contributions of this work can be summarized as follows:

1. We propose to Regularize, Expand and Compress (REC) for CL, which automatically expands the network capacity for continuous learning a new task with fewer parameters than human-designed architectures. The final model is a non-expensive

Table 7.1: Comparisons of the Lifelong Learning Approaches for Overcoming Catastrophic Forgetting. EWC: Elastic Weight Consolidation (Kirkpatrick *et al.*, 2017); DEN: Dynamically Expandable Network (Yoon *et al.*, 2017); LwF: Learning without Forgetting (Li and Hoiem, 2017); GEM: Gradient of Episodic Memory (Lopez-Paz *et al.*, 2017); PGN: Progressive Neural Network (Rusu *et al.*, 2016) and Our Algorithm REC.

	EWC	DEN	LwF	GEM	PGN	REC
Overcome catastrophic forgetting	✓	✓	✓	✓	✓	✓
No memory growth	✓		✓	✓		✓
No exemplar	✓	✓	✓		✓	✓
Can expand network capacity		✓			✓	✓
AutoML ability						✓

model but the performance is significantly enhanced by network expanding procedure.

2. To overcome the catastrophic forgetting of the previously learned tasks, we propose Regularized Weight Consolidation (RWC) — it identifies and retrains the discriminative subset of weights by exploiting inherent correlations among the tasks and trains the newly added layer as a task-specific layer for the new task.

3. Furthermore, REC applies an economical and efficient network transformation on arrival of the new task, which is advantageous over traditional AutoML frameworks, which discards the trained network and searching the architecture from scratch.

7.2 Related Work

7.2.1 Overcoming Catastrophic Forgetting

Recently, a lot of lifelong learning methods were proposed to address the catastrophic forgetting problem. The first group of methods uses regularized learning. Elastic Weight Consolidation (EWC) (Kirkpatrick *et al.*, 2017) shows that task-specific synaptic consolidation may overcome catastrophic forgetting in neural networks and observes the important weights for the previous tasks and selectively ad-

justs the plasticity of the weights. Inspired by EWC, Schwarz *et al.* (2018) propose online EWC, which enlarges the EWC scalability by limiting the regularization term computational cost when the number of tasks increases. Synaptic Intelligence (Zenke *et al.*, 2017) computes an online importance measure along an entire learning trajectory, which is similar to EWC. Rotate-EWC (Liu *et al.*, 2018) (REWC) is a modified version of EWC — it approximately diagonalizes the Fisher information matrix of the network parameters that compute the factorized rotation of the parameter space used in conjunction with EWC.

The second group of the strategies is associated with learning task-specific parameters. Learning without forgetting (LwF) (Li and Hoiem, 2017) leverages distillation regularization on the new tasks — the soft labels of previously learned tasks are enforced to be similar to the network with the current task by using knowledge distillation (Hinton *et al.*, 2015). Less-forgetful learning (Jung *et al.*, 2017) is proposed to regularize the L_2 distance between the final hidden activations and the old tasks' parameters for preserving the old task feature mappings.

The third group of methods expands the network capacity. Progressive neural network (PGN) (Rusu *et al.*, 2016) is proposed to block any changes to the pre-trained network models on previously learned tasks and expands the network architecture by allocating sub-networks with the fixed capacity to be trained with the new information. PathNet (Fernando *et al.*, 2017) uses agents embedding into a neural network to find which parts of the network can be reused for learning new tasks and freezes task-relevant paths for avoiding catastrophic forgetting. Dynamically expanding network (DEN) (Yoon *et al.*, 2017) increases the number of trainable parameters to continually learn new tasks and dynamically selects neurons to retrain or expand neuron capacity by using group sparse regularization.

The other family of the methods uses episodic memory, where the previously learned task samples are stored to effectively recall the experience in the past. Gradient of Episodic Memory (GEM) (Lopez-Paz *et al.*, 2017) performs positive forward transfer, minimizes negative backward transfer to previously learned tasks and learns the subset of correlations to a set of tasks without using task descriptors. Incremental Classifier and Representation Learning (iCaRL) (Rebuffi *et al.*, 2017) combines classification loss on new tasks and distillation loss on previously learned tasks with a K-nearest neighbor classifier and selects the exemplars for each task by letting the embeddings of the selected samples closer to the center point of each class. Table 7.1 shows the multiple merits of REC, comparing with previous researches in this area.

7.2.2 AutoML and Knowledge Distillation

There are many works on AutoML to improve the performance of deep neural networks (Zoph and Le, 2016; Pham *et al.*, 2018; Cai *et al.*, 2018). Neural Architecture Search (NAS) (Zoph and Le, 2016) searches the transferable network blocks via reinforcement learning and outperforms many manually designed network architecture. ENAS (Pham *et al.*, 2018) uses a controller to discover network architectures by searching an optimal subgraph within a large computational graph and shares parameters among child models to enable efficient NAS. EAS (Cai *et al.*, 2018) efficiently explores network architecture via network transformation (Chen *et al.*, 2015) which is a functionality preserving method to expand the architecture with a fixed number of units or filters.

Besides, Knowledge distillation (KD) (Hinton *et al.*, 2015) is also very related to our work. KD is widely used to compress a network with a different architecture that approximates the original network where knowledge is transferred from a large teacher network to a small student network. The student network is trained with KD loss –a

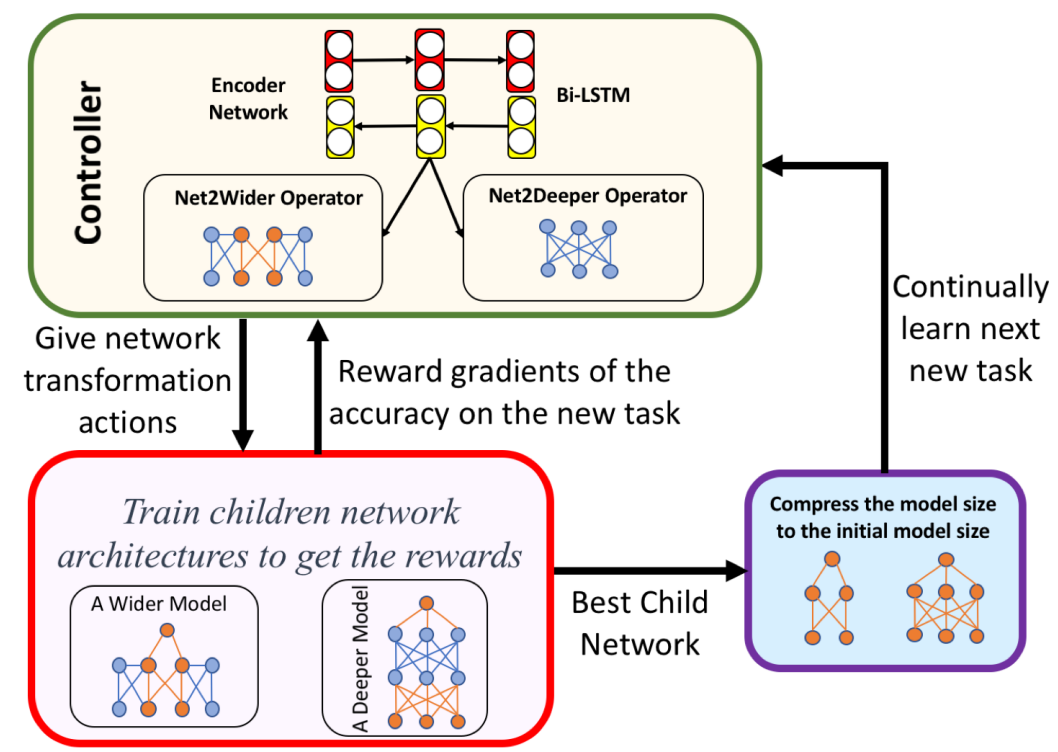


Figure 7.2: Illustration of Our CL framework. REC First Searches the Best Child Network by RWC with Net2Deeper and Net2Wider Operators in the Controller for a New Coming Task, Then Compresses the Expanded Network to the Same Size as the Initial Model and Continually Learns Next New Task.

modified cross-entropy loss– that ensures the teacher network and student network are similar. In our work, we adopt the KD to compress the expanded network after learning each new task.

7.3 Method

Fig. 7.2 is an overview of our NonExpensive AutoML framework REC for CL with three components.

7.3.1 Problem Definition and Overview

We define the continual learning problem as follows — there will be an unknown number of tasks with unknown distributions, arriving in sequence. Our goal is to learn

a deep model in such a continual learning scenario without catastrophic forgetting. For the evaluation protocol, we report the classification accuracy on each of previous $T-1$ tasks and the current task T after training on the T-th task. Given a sequence of T tasks, task at time point $t = 1, 2, \cdots, T$ with N_t images comes with dataset $\mathbf{D}_t = \{\mathbf{x}_i^t, y_i^t\}_{i=1}^{N_t}$. Specifically, for task t, $y_i^t \in \{1, ..., K\}$ is the label for the i-th sample $\mathbf{x}_i^t \in \mathbb{R}^{d_t}$ in task t. We denote the training data matrix by $\mathbf{X}^t$ for $\mathbf{D}_t$, i.e., $\mathbf{X}^t = (\mathbf{x}_1^t, \cdots, \mathbf{x}_{N_t}^t)$. When the dataset of task t comes, all the previous training datasets $\mathbf{D}_1, \cdots, \mathbf{D}_{t-1}$ are not available any more, but the deep model parameter $\theta^{t-1} = \{\theta_l^{t-1}\}_{l=1}^L$ can be accessed. The continual learning problem at time point t when given data $\mathbf{D}_t$ can be defined as solving the following problem:

$$\min_{\theta_t} \mathcal{F}(\theta_t | \theta_{t-1}, \mathbf{D}_t), t = 1, \cdots, T \tag{7.1}$$

where $\mathcal{F}$ is the loss function of solving θ^t, θ^t is the parameter for task t.

Kirkpatrick *et al.* (2017) proposed EWC that consists of a quadratic penalty on the difference between the parameter θ^t and θ^{t-1} to slow down the catastrophic forgetting for previously learned tasks. The posterior distribution $p(\theta^t|\mathbf{D}_t)$ is used to describe the problem by the Bayes' rule.

$$\log p(\theta^t|\mathbf{D}_t) = \log p(\mathbf{D}_t|\theta^t) + \log p(\theta^t|\mathbf{D}_{t-1}) - \log p(\mathbf{D}_t), \tag{7.2}$$

where the posterior probability $\log p(\theta^t|\mathbf{D}_{t-1})$ embeds all the information from task $t-1$. However, the problem (7.2) is intractable so that EWC approximates it as a Gaussian distribution with mean of parameter $\bar{\theta}^{t-1}$ and a diagonal I of the Fisher Information matrix $\mathbb{F}$. The matrix $\mathbb{F}$ is computed by $\mathbb{F}_i = I(\theta^t)_{ii} = E_x[(\frac{\partial}{\partial \theta_i^t} \log p(\mathbf{D}_t|\theta^t))^2|\theta^t]$. Therefore, the problem of EWC on task t can be written as follows:

$$\min_{\theta^t} \quad \mathcal{F}_t(\theta^t) + \frac{\lambda}{2} \sum_i \mathbb{F}_i(\theta_i^t - \bar{\theta}_i^{t-1})^2, \tag{7.3}$$

where $\mathcal{F}_t$ is the loss function for task t, λ denotes how important the task $t-1$ is compared to the task t and i labels each weight of the parameter θ.

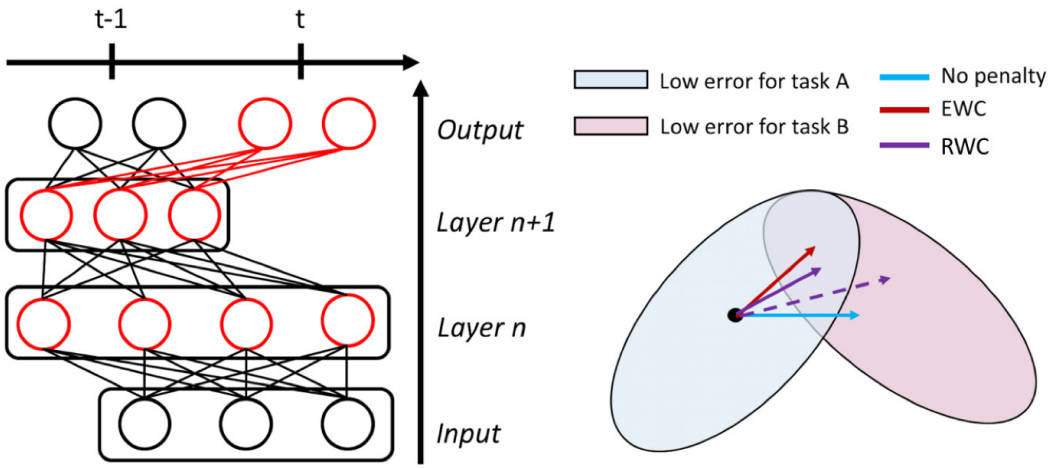


Figure 7.3: RWC Retrains the Entire Network Learned on Previous Tasks while Regularizing It to Prevent Forgetting from the Original Model. RWC (Purple Solid Line) Learns Better Parameter Representations to Overcome Catastrophic Forgetting by Studying MTL with the Sparsity-Inducing Norm (Purple Dash Line) and EWC (Red Line).

7.3.2 Regularized Weight Consolidation

The main problem of EWC is that EWC only enforces task t close to task $t-1$, but ignores the inherent correlations between task $t-1$ and task t and such relationship might potentially help overcome catastrophic forgetting on the task $t-1$. Learning multiple related tasks jointly can improve performance relative to learning each task separately, when the tasks are related — this idea is incorporated into Multi-Task Learning (MTL) (Evgeniou and Pontil, 2004). It has been commonly used to obtain better generalization performance than learning each task individually. We regularized Eq. 7.3 via MTL and propose a new objective function Eq. 7.4 to overcome catastrophic forgetting from multiple tasks simultaneously:

$$\min_{\theta^t} \mathcal{F}_t(\theta^t) + \frac{\lambda}{2}\sum_i \mathbb{F}_i(\theta_i^t - \bar{\theta}_i^{t-1})^2 + \lambda_2 |||[\theta^t; \theta^{t-1}]|||_{2,1}, \tag{7.4}$$

where λ_2 is the non-negative regularization parameter and $|||[\theta^t; \theta^{t-1}]|||_{2,1} = ||||\theta^t||_2, ||\theta^{t-1}||_2||_1$ is the $l_{2,1}$-norm regularization to learn the related representations and capture the common subset of relevant parameters from each layer for task $t-1$

and task t.

Specifically, we further consider some important parameters which have better representation power to a subset of tasks. The sparsity-inducing norm (Gong *et al.*, 2012) has been studied in this chapter to select such discriminative parameter subset by incorporating inherent correlations among multiple tasks. To this end, the l_1 sparse norm is imposed to learn the new task-specific parameters while learning task relatedness among multiple tasks. Therefore, the objective function for task t becomes:

$$\min_{\theta^t} \quad \mathcal{F}_t(\theta^t) + \frac{\lambda}{2}\sum_i \mathbb{F}_i(\theta_i^t - \bar{\theta}_i^{t-1})^2 + \lambda_2||[\theta^t; \theta^{t-1}]||_{2,1} + \lambda_3||\theta^t||_1, \tag{7.5}$$

where λ_3 is the non-negative regularization parameter. We call our algorithm Regularized Weight Consolidation (RWC) and Fig. 7.3 shows the geometric illustration of RWC.

7.3.3 NonExpansive Continual Learning

RWC is a regularization-based CL, it might be needed to expand the network if the task is very different from the existing ones or the network capacity is not sufficient when more and more newly coming tasks. Due to human experts usually find a sub-optimal solution, this encourages us to propose AutoML based network expanding method for CL to find a global optimal solution. We name it Regularize, Expand, Compress (REC) and summarize the steps in Algorithm 9 and the details of expanding network are outlined in Algorithm 10.

We consider net2wider and net2deeper operators (Chen *et al.*, 2015) to expand the network capacity. The net2wider network transformation function is as follows:

Algorithm 9: Regularize, Expand and Compress (REC)

Input : Dataset $\mathbf{D}_1, \cdots, \mathbf{D}_T, \lambda, \lambda_1, \lambda_2$

Output: θ_c^T

```
begin
    for t = 1 → T do
        if t = 1 then
            Train an initial network with weights θ^1 by using Eq. 7.1.
        else
            Search a best child network θ^t by Alg. 10 with Eq. 7.8.
            Compress θ^t to the same model size as θ^1 by Eq. 7.10 and use θ_c^t for
             next task.
```

$$\pi_{wider}(j) = \begin{cases} j & j \leq O_l, \\ random\ sample\ from\ \{1, ..., O_l\} & j > O_l, \end{cases} \tag{7.6}$$

where O_l represents the outputs of the original layer l. And the net2deeper network transformation function is

$$\gamma(\pi_{deeper}(j)) = \gamma(j) \quad \forall j. \tag{7.7}$$

where the constraint γ holds for the rectified linear activation. We learn a meta-controller to generate network transformation actions (Eq. 7.6 and Eq. 7.7) when given the initial network architecture. Specifically, we use an encoder network (Cai *et al.*, 2018), which is implemented with an input embedding layer and a bidirectional recurrent neural network (Schuster and Paliwal, 1997), to learn a low-dimensional representation of the initial network and be embedded into different operators to generate different network transformation actions. Besides, we use a shared sigmoid classifier to make the Net2Wider decision according to the hidden state of the layer

learned by the bidirectional encoder network (Cai *et al.*, 2018) and the wider network can be further combined with a Net2Deeper operator.

We integrate RWC (Eq. 7.5) into the AutoML framework as the loss function for CL settings. After we learning the network θ^{t-1} on the data $\mathbf{D}^{t-1}$, we will automatically search the best child network θ^t for task t among all the generated children networks $\theta_1^t, ..., \theta_m^t$ (m is the number of children networks). The network expansion will be finished by Net2wider and Net2Deeper operators when it is necessary to expand the network. If the controller decides to expand the network, the newly added layer will not have the previous tasks' Fisher Information. We consider the newly added layer as a new task-specific layer, l_1 regularization is adopted to promote sparsity in the new weight so that each neuron only connected with few neurons in the layer below. This will efficiently learn the best representation for the new task while reducing the computation overheads. The modified RWC in the network expansion scenario as follows:

$$\min_{\theta^t} \quad \mathcal{F}_t(\theta^t) + \frac{\lambda}{2} \sum_{\substack{i \neq deeper \\ i \neq wider}} \mathbb{F}_i(\theta_i^t - \bar{\theta}_i^{t-1})^2 + \lambda_2 ||[\theta^t; \theta^{t-1}]||_{2,1} + \lambda_3 ||\theta^t_{\substack{i=deeper \\ i=wider}}||_1, \tag{7.8}$$

where the subscript *deeper* and *wider* refer to the newly added layer in task t.

After the controller generates the child network, the child network will achieve an accuracy A_{val} on the validation set of task t and this will be used as the reward signal R^t to update the controller. We maximize the expected reward to find the optimal child network. The empirical approximation of our AutoML REINFORCE rule (Sutton *et al.*, 2000) as follows:

$$\frac{1}{m} \sum_{i=1}^{m} \sum_{s=1}^{S} \bigtriangledown_C \log P(a_s | a_1, \cdots, a_{s-1}; C) R_i^t, \tag{7.9}$$

where m is the number of children networks that the controller C samples and a_s and g_s represents the action and state of predicting s-th hyperparameter to design a child

network architecture, respectively. In Alg. 10, T is the transition function. Since R^t is non-differentiable, we use policy gradient to update the controller. We use a non-linear transformation $tan(A_{val} \times \pi/2)$ on validation set of task t as done in (Cai *et al.*, 2018) and use the transformed value as the reward. We also use an exponential moving average of previous rewards with a decay of 0.95 to reduce the variance. To balance the old task and new task knowledge, we set maximum expanding layers are 2 and 3 on net2wider and net2deeper operators, respectively.

If the network keeps expanding as more and more tasks will be given, the model will suffer the inefficient problem and have extra memory cost. Thus, the model compression technique is needed to reduce the memory cost and receive a nonexpansive model. Here, we use soft-label (the logits) as knowledge distillation (KD) (Hinton *et al.*, 2015) instead of the hard labels to train the student model. To be noticed, the θ^t has learned the knowledge of new task t and old tasks $1, \cdots, t-1$. The compressed model θ_c^t will have the similar performance as θ^t and it is not really necessary learning the parameter of θ^{t-1} again. We follow Ba and Caruana (2014) that the student model is trained to minimize the mean of the l_2 loss on the training data $\{\mathbf{x}_i^t, z_i^t\}_{i=1}^{N^t}$, where z_i^t is the logits of the child model θ^t i-th training sample. We compress the θ^t to the same size model as θ^1 as long as we expand the network, the KD loss is listed below:

$$\min_{\theta_c^t} \mathcal{F}_{kd}(f(\mathbf{x}^t; \theta_c^t), \mathbf{z}^t) = \frac{1}{N^t} \sum_i ||f(\mathbf{x}_i^t; \theta_c^t) - z_i^t||_2^2, \tag{7.10}$$

where θ_c^t is the weights of the student network and $f(\mathbf{x}_i^t; \theta_c^t)$ is the prediction of task t i-th training sample.

The final student network θ_c^t is trained to convergence with hard and soft labels by the following loss function:

Algorithm 10: Automatically Network Transformation

Input : Dataset $\mathbf{D}_t$, θ^{t-1}

Output: The best expended model θ^t

begin
  for $i = 1 \rightarrow m$ **do**
    for $s = 1 \rightarrow S$ **do**
      $a_s \leftarrow \pi_{deeper}(g_{s-1}; \theta^{t-1}_{deeper})$ or $\pi_{wider}(g_{s-1}; \theta^{t-1}_{wider})$
      $g_s \leftarrow T(g_{s-1}, a_s)$
      $\theta^t \leftarrow \theta^t_{newLayer}$
    $R_i \leftarrow tanh(A^t_i(g_S) \times \pi/2)$
    $\theta^t_i \leftarrow \bigtriangledown_{\theta^t_{i-1}} J(\theta^t_{i-1})$

$$\min_{\theta^t_c} \mathcal{F}(f(\mathbf{x}^t; \theta^t_c), \mathbf{y}^t) + \mathcal{F}_{kd}(f(\mathbf{x}^t; \theta^t_c), \mathbf{z}^t), \tag{7.11}$$

where $\mathcal{F}$ is the loss function (cross-entropy) for training with ground truth $\mathbf{y}^t$ of task t.

7.4 Experiments

7.4.1 Experimental Settings

Datasets. We evaluate our algorithm on most commonly used datasets for CL. We list them as follows:

-MNIST-permutation: MNIST (LeCun, 1998) is used as the most common datasets among all lifelong learning works, which consists of ten handwritten digits classes with 60,000/10,000 training and testing examples. One way to create the datasets for multiple tasks is randomly permuting the pixels by a fixed permuta-

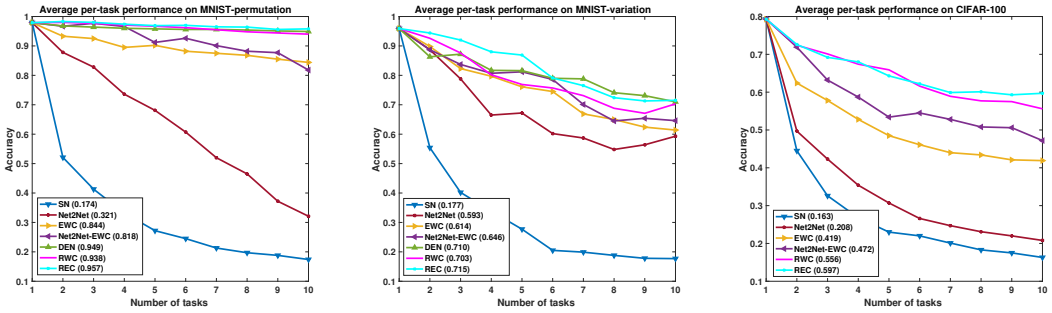


Figure 7.4: The Experimental Results of Continual Training on MNIST-permutation, MNIST-variation and CIFAR-100 Datasets. We Report the Average per-Task Performance (Accuracy) of the Models over $T = 10$ Task. The Numbers in the Legend Represent Average per-Task Performance after the Model Has Finished Learning Task t.

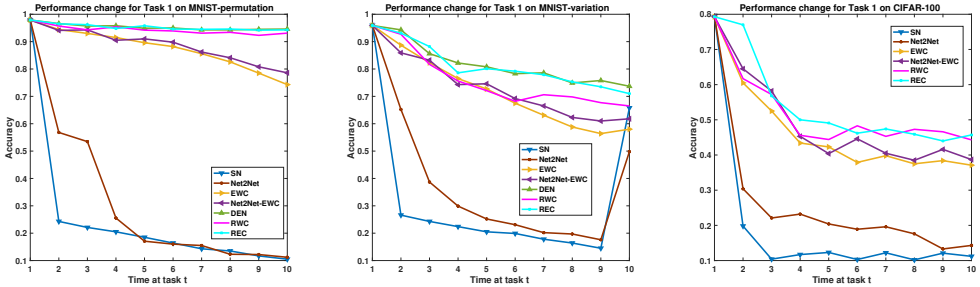


Figure 7.5: Forgetting Experiment for Task 1 on MNIST-permutation, MNIST-variation and CIFAR-100 Datasets. We Report the Accuracy of Different Models on Task $t = 1$ at Each Training Stage to See How the Model Performance Changes Over Time for All Datasets.

tion (Kirkpatrick *et al.*, 2017) so that the input distribution for each task is unrelated.

-MNIST-Variation: MNIST-variation (LeCun, 1998) dataset rotates the MNIST dataset by a fixed angle between 0 to 180 degrees for each different task. We use $180/T$ as the fixed angle to create T tasks.

-CIFAR-100: CIFAR-100 (Krizhevsky and Hinton, 2009) dataset contains 60,000 32×32 color images in 100 object classes. Each class has 500/100 images for training and testing. We consider each task with a set of classes, it contains $100/T$ classes when there are T tasks. Different from MNIST-permutation dataset, the input distributions are similar for all tasks but the output distributions for each task are different.

-CUB-200: CUB-200 (Wah *et al.*, 2011) is a fine-grained image classification benchmark, we use CUB-200-2011 version in this work. It contains 11,788 images of 200 types of birds with 5,994/5,794 for training and testing. Each image has detailed annotations and a bounding box. We crop the bounding boxes from the original images and resize them to 224×224. We use the same way to create multiple tasks as CIFAR-100 dataset.

For the first three datasets, we choose $T = 10$ tasks. Since the fine-grained CUB-200 dataset is more challenging than others, we set $T = 4$ tasks to show better comparisons on lifelong learning. For all datasets, we use 0.1 ratio to split validation set and the model observes the tasks in sequence. We generate multiple tasks for each dataset first and all comparison methods then use the same task order and the same categories within the task for fair comparisons.

Base network settings. For two MNIST datasets, we use a two-layer fully-connected neural network of 100-100 units with ReLU activations as our initial network. For CIFAR-100 dataset, we use a modified version of AlexNet (Krizhevsky *et al.*, 2012) which has five convolutional layers (64-128-256-256-128 depth with 5×5 filter size), and three fully-connected layers (384-192-100 neurons at each layer) and the standard data augmentation is used in this dataset. For CUB-200 dataset, we use a pre-trained VGG-16 (Simonyan and Zisserman, 2014) model from ImageNet (Deng *et al.*, 2009) and fine-tune it on the CUB-200 data for better initialization. We follow the setting of Liu *et al.* (2018), which adds a global pooling layer after the final convolutional layer of the VGG-16. The fully-connected layers are changed to 512-512 and the size of the output layer is the number of classes in each task. All models and algorithms are implemented using Tensorflow (Abadi *et al.*, 2016) library.

Comparison methods. We compare our algorithm with six other methods: 1) SN: A single network trained across all tasks. 2) Net2Net (Chen *et al.*, 2015):

Network expanding by Net2Net (Chen *et al.*, 2015) on new task. 3) EWC (Kirkpatrick *et al.*, 2017): A deep network trained with elastic weight consolidation. 4) Net2Net-EWC: Network expanding by Net2Net (Chen *et al.*, 2015) with elastic weight consolidation (Kirkpatrick *et al.*, 2017) when learning new task. 5) DEN (Yoon *et al.*, 2017): Dynamically expandable network. 6) REWC (Liu *et al.*, 2018): Rotate Elastic Weight Consolidation. 7) RWC: A deep network trained with regularized weight consolidation. 8)REC: Regularize, Expand and Compress.

Hyperparameter settings. All hyper-parameters in RWC are optimized using a grid-search and the best results for each model are reported. For two MNIST datasets, the SGD optimizer is used with a learning rate of 0.001 and we set batch size of 256 with 8 epochs, $\lambda_1 = 2$, $\lambda_2 = 0.0001$ and $\lambda_3 = 0.001$ in all experiments. For CIFAR-100 dataset, we use SGD optimizer with momentum parameter of 0.9, learning rate of 0.01, batch size of 128 with 20 epochs, $\lambda_1 = 10$, $\lambda_2 = 0.015$ and $\lambda_3 = 0.0001$. For CUB dataset, the Adam optimizer is used with a learning rate of 0.001, batch size of 32 and 50 epochs, $\lambda_1 = 100$, $\lambda_2 = 0.001$ and $\lambda_3 = 0.005$. For network transformation based AutoML experimental settings, we followed the training details of Cai *et al.* (2018).

7.4.2 Experimental Results

We evaluate our methods from both model accuracy and model complexity, where we measure the model size at the end of the training process.

Comparisons of the model performance. We report the average per-task accuracy of MNIST-permutation, MNIST-variation and CIFAR-100 datasets when $T = 10$ in Fig. 7.4 and average the results over five runs. Overall, REC outperforms all comparison methods and overcomes catastrophic forgetting especially on the later tasks (after task 5). We can observe that the regularization based net-

Table 7.2: Comparisons of the Model Size and the Average Task Accuracy after Training 10 Tasks on MNIST-permutation Dataset. $\#W(1)$: Total Parameters of Task 1. $\#W(10)$: Total Parameters of Task 10. ACC (10): Average per-Task Accuracy after Task 10.

Methods	#W (1)	#W (10)	ACC (10)
SN	0.01M	0.01M	17.4%
Net2Net	0.01M	0.02M	32.1%
EWC	0.01M	0.01M	84.4%
Net2Net-EWC	0.01M	0.02M	81.8%
DEN	0.01M	0.14M	94.9%
RWC	0.01M	0.01M	93.8%
REC	**0.01M**	**0.01M**	**95.7%**

work (EWC, RWC) has worse performance than expandable networks (DEN, REC), which shows that selectively expand networks help improve the performance by a large margin. Specifically, REC performs better than DEN on two MNIST datasets and RWC performs similarly with DEN on MNIST-permutation dataset while using fewer parameters. We also observe that directly apply Net2Net (Chen *et al.*, 2015) on lifelong learning does not perform well since it forgets the old tasks' knowledge as finetuning (SN), but adding EWC as the loss function can help enhance the old tasks' performance on Net2Net. REC has better performance than Net2Net-EWC, because we consider the new task-specific parameters and the discriminative common subset between the old tasks and the new one.

We also evaluate the catastrophic forgetting over time on the earliest task, Fig. 7.5 shows the test accuracy of the first task throughout the whole lifelong learning process on MNIST-permutation, MNIST-variation and CIFAR-100 datasets. It shows that our methods (RWC and REC) overcome forgetting on old tasks compared with all other methods on MNIST-permutation and CIFAR-100 datasets. It is worth noting

Table 7.3: Comparisons of the Model Size and the Average Task Accuracy after Training 10 Tasks on CIFAR-100 Dataset. $\#W(1)$: Total Parameters of Task 1. $\#W(10)$: Total Parameters of Task 10. ACC (10): Average per-Task Accuracy after Task 10.

Methods	#W (1)	#W (10)	ACC (10)
SN	4M	4M	16.3%
Net2Net	4M	6.3M	20.8%
EWC	4M	4M	41.9%
Net2Net-EWC	4M	7.4M	47.2%
RWC	4M	4M	55.6%
REC	**4M**	**4M**	**59.7%**

that DEN performs slightly better than our method on task 1 after learning later tasks on MNIST-variation dataset due to they selectively expands network for the new task, it will give a bias towards to the earliest task. Our REC is a nonexpensive network and our overall average per-task performance is better than DEN, which shows that our method has better performance on later learned tasks and achieve a more balanced performance when learning sequential tasks in the temporal dimension comparing with DEN. Besides, we have an interesting founding on MNIST-variation dataset, the SN and Net2Net has irregular performance on task 1 after learning task 10, it is due to the task 10 is the upside-down flipped image of task 1 and such flip gives benefit on some digits such as '1','0','8'. And SN and Net2Net forget too much task 1' knowledge after learning task 9, they only can keep the most recently learned task knowledge when they learn task 10 comparing with EWC, RWC and REC and this causes the irregular performance.

Comparisons of the model complexity. Table 7.2 and Table 7.3 report the comparisons of the model size and the average per-task performance after training $T = 10$ tasks of different approaches on MNIST-permutation and CIFAR-100 datasets,

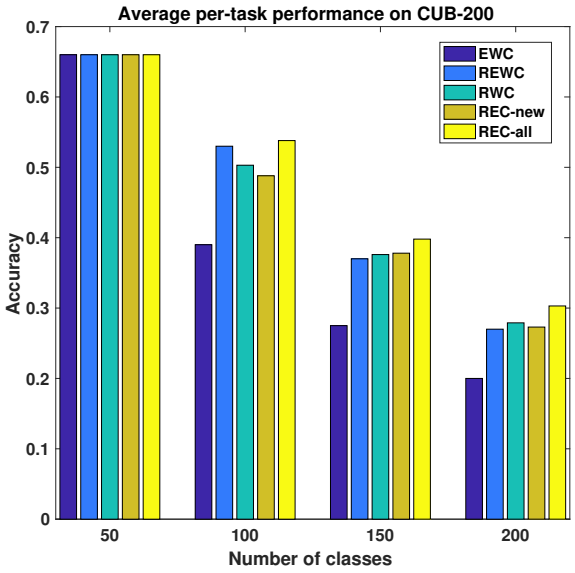


Figure 7.6: Comparison Results with EWC and REWC on CUB-200 Dataset When $T = 4$.

respectively. Overall, REC performs similarly or better than all other approaches with smaller model size. We observe that DEN performs better than RWC and worse than REC on MNIST-permutation dataset, but it has 1.4X network expansion comparing with ours. For CIFAR-100 dataset, We compute our AUROC after learning $T = 10$ tasks, REC can achieve 0.887 comparing with DEN (0.923), however, our model size is 50% of DEN's model. Besides, we notice that DEN involves 7 hyperparameters and very sensitive to them, we slightly change one of them from 10^{-3} to 10^{-2}, the result becomes 0.8907 on MNIST-permutation dataset. Our method only has three hyperparameters and it needs much less expert tuning comparing with DEN. Training times is a limitation of the current version of REC, since REC is a reinforcement learning based algorithm, a varies number of trails are needed and this results in more training time than other methods. We will improve the training efficiency of our work in the future. Besides, we did not consider complexity network structures (e.g. ResNet (He *et al.*, 2016b), DenseNet (Huang *et al.*, 2017)), we will extend the current work to more network architectures in the future.

Table 7.4: Comparison Results of Average per-Task Accuracy after Training Task 10 on MNIST-permutation Dataset.

Method	EWC	EWC+l_1	EWC+$l_{2,1}$	RWC
ACC(10)	84.4%	87.7%	88.5%	94.0%

Results on CUB-200 dataset. Fig. 7.6 shows the comparison results when $T = 4$ on CUB-200 dataset with EWC (Kirkpatrick *et al.*, 2017) and REWC (Liu *et al.*, 2018). It shows that RWC has comparable results with REWC, RWC has better performance on task 3 and task 4 while has worse performance on task 2. We test REC with only new task validation set (REC-new), which has similar results as RWC on later tasks. This might be caused by using only new task validation set is not sufficient to compute the rewards on a more subtle dataset. We hypothesis the exemplars from old tasks will help improve the nonexpansive AutoML system's performance. Thus, we use the validation sets of all learned tasks to compute the rewards and report the results (REC-all) in Fig. 7.6. The results show that exemplars from old tasks help improve the performance of AutoML based algorithm and we will investigate the relationship between the number of exemplars and the performance of REC in our future work.

Ablation study on each component in RWC. We study how the different components used in RWC affect the final performance of lifelong learning. We report the average per-task accuracy after training task 10 on MNIST-permutation of different strategies *EWC*, *EWC with l_1-norm only*, *EWC with l_2-norm only* and *RWC* in Table 7.4. It shows that $l_{2,1}$-norm has a stronger effect of the performance than l_1-norm while our method RWC outperforms the single regularization strategies, which demonstrates the meaningful and useful of our method by studying common weights subset with discriminative new task parameters.

7.5 Summary

In this work, we develop a regularized continual learning framework via nonexpansive AutoML (REC). REC is achieved at two stages: continually network expansion and model compression. To overcome catastrophic forgetting, we propose RWC. We achieve better accuracy and smaller model size than other CL methods on four datasets.

Model compression is an optional stage for the current work with a trade-off between the compressed model and the original model. REC is our initial work for overcoming catastrophic forgetting and we will speed-up the hyperparameter optimization (Hinz *et al.*, 2018) in our future work. The AutoML training time is another limitation with REC, however it can be further improved by optimality tightening (He *et al.*, 2016a) or parallelization (Zoph and Le, 2016) or similar approaches for reducing the training time. We plan to reduce the training complexity in our future work.

Chapter 8

CONCLUSIONS AND FUTURE WORK

In this chapter, I summarize the major contributions of this dissertation. Moreover, I will discuss some of my current ongoing work and possible future research directions.

8.1 Summary

In this dissertation, I have proposed three groups of algorithms to address the data scarcity problem that is one of the major bottlenecks for building a robust machine learning system.

I first proposed two unsupervised conventional machine learning algorithms, hyperbolic stochastic coding (HSC), and multi-resemble multi-target low-rank coding (MMLC), to solve the incomplete data and missing ground truth problem.

As neural networks achieve cutting-edge performance over the conventional machine learning methods, I proposed a temporally adaptive sparse network (TaDsNet) and a deep domain adaptation multi-ROIs learning network (DDAML) to leverage the benefit of deep learning. Meanwhile, we transfer the rich knowledge from a large-amount labeled source dataset for addressing the data scarcity problem.

Aiming to build an artificial general intelligent model, Lifelong machine learning (LML) plays the role of learning as humans do, i.e., retaining the results learned in the past, abstracting knowledge from them, and using the knowledge to help model future learning. Incorporate this capability and make it versatile, holistic, and intelligent, I proposed deep multi-task weight consolidation (dMopWC) and non-expanded deep continual learning regularize, expand and compress (REC) algorithms to accumulate knowledge continuously and drastically reduce data requirements in a variety

of domains. The rationale is that when faced with a new situation, we humans use our previous experience and knowledge to help deal with the new situation, and such ability will make major progress in the AI revolution.

8.2 Ongoing Work

In my current ongoing work, I propose alternative methods of supervised learning that do not require direct labels. Intuitively, although we do not know what the labels are, we might identify various properties they should satisfy. The key idea is to formulate these properties as objectives for supervising the target tasks. We show that this kind of 'self-supervision' on how the output behaves, rather than what it is, turns out to be surprisingly effective in learning a variety of vision tasks. For instant, not all frames of a video capturing a dynamic human action are of equal importance. Few frames, often best summarize the gist of the human action much more effectively than others.

My latest work presents an original approach to selecting such frames from arbitrary videos without the supervision of the action labels or key frame labels. We propose a novel deep neural network architecture for selecting key frame that is trained by optimizing a combination of losses. The key frames should be able to represent the underlying action well with weakly supervision by knowledge distillation from a given teacher model. The key frame should self indented in the action videos their difference by their own properties to supervise itself learning. Our model is unsupervised in the sense that it does not require any training data containing labeled key frames or labeled action videos. We argue that the proposed method only requires few-shot labeled data to train, and it is able to self-supervised learning the information it needs and has the ability to generalize to datasets not seen during training due to self-learning ability.

8.3 Future Directions

Supervised learning and current deep neural network can solve the image-related task very well while enough labeled data are given. The problem is that the high performance of the deep models relies on a massive amount of labeled data. However, in reality, to obtain the manual labels is extremely expensive, and the quality of the annotation might differ from each other. Besides, the size of the dataset is hard to be scaled up.

Unsupervised learning usually works much less efficiently than supervised learning. But it is wasteful if we ignore the labeled dataset, even the amount of unlabelled data is substantially more than the limited amount of human annotation data.

I believe the next AI revolution will focus on getting labels for free on the unlabelled data and train the unsupervised neural network in a supervised manner. We can achieve this by formulating various properties of the data as some supervision and treat it as a supervised learning task to predict the information. In this way, we can fully utilize the limited amount of labeled data and a vast amount of unlabelled data. This is known as self-supervised learning.

In the future, I will continue working on developing self-supervised learning methods so that we can train a model in online fashion with only a few-shot labeled data toward robust machine learning.

Abadi, M. *et al.*, "Tensorflow: a system for large-scale machine learning.", in "OSDI", vol. 16, pp. 265–283 (2016).

Argyriou, A., T. Evgeniou and M. Pontil, "Convex multi-task feature learning", Machine Learning **73**, 3, 243–272 (2008).

Ashburner, J., C. Hutton, R. Frackowiak, I. Johnsrude, C. Price and K. Friston, "Identifying global anatomical differences: deformation-based morphometry", Hum Brain Mapp **6**, 5-6, 348–357 (1998).

Ba, J. and R. Caruana, "Do deep nets really need to be deep?", in "Advances in neural information processing systems", pp. 2654–2662 (2014).

Balasubramanian, K., K. Yu and G. Lebanon, "Smooth sparse coding via marginal regression for learning sparse representations", in "Proceedings of the 30th International Conference on Machine Learning (ICML-13)", pp. 289–297 (2013).

Beach, T. G., S. E. Monsell, L. E. Phillips and W. Kukull, "Accuracy of the clinical diagnosis of Alzheimer disease at National Institute on Aging Alzheimer Disease Centers, 2005-2010", J. Neuropathol. Exp. Neurol. **71**, 4, 266–273 (2012).

Ben Ahmed, O., M. Mizotin, J. Benois-Pineau, M. Allard, G. Catheline and C. Ben Amar, "Alzheimer's disease diagnosis on structural MR images using circular harmonic functions descriptors on hippocampus and posterior cingulate cortex", Comput Med Imaging Graph **44**, 13–25 (2015).

Bickel, P. J., Y. Ritov and A. B. Tsybakov, "Simultaneous analysis of lasso and dantzig selector", The Annals of Statistics pp. 1705–1732 (2009).

Bottou, L., "Online learning and stochastic approximation", Online Learning and Neural Networks, Cambridge University Press, Cambridge, UK (1998).

Boureau, Y.-L., J. Ponce and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition", in "Proceedings of the 27th international conference on machine learning (ICML-10)", pp. 111–118 (2010).

Boyd, S. and L. Vandenberghe, *Convex optimization* (Cambridge university press, 2004).

Brand, L., K. Nichols, H. Wang, L. Shen and H. Huang, "Joint Multi-Modal Longitudinal Regression and Classification for Alzheimer's Disease Prediction", IEEE Trans Med Imaging (2019).

Bredies, K. and D. A. Lorenz, "Linear convergence of iterative soft-thresholding", Journal of Fourier Analysis and Applications **14**, 5-6, 813–837 (2008).

Cai, H., T. Chen, W. Zhang, Y. Yu and J. Wang, "Efficient architecture search by network transformation", in "AAAI", (2018).

Canutescu, A. A. and R. L. Dunbrack, "Cyclic coordinate descent: A robotics algorithm for protein loop closure", Protein science **12**, 5, 963–972 (2003).

Cao, J., K. Worsley, C. Liu, L. Collins and A. Evans, "New statistical results for the detection of brain structural and functional change using random field theory", NeuroImage **5**, 4, 512 (1997).

Chen, J., L. Tang, J. Liu and J. Ye, "A convex formulation for learning shared structures from multiple tasks", in "Proceedings of the 26th Annual ICML", pp. 137–144 (ACM, 2009).

Chen, T., I. Goodfellow and J. Shlens, "Net2net: Accelerating learning via knowledge transfer", arXiv preprint arXiv:1511.05641 (2015).

Chou, Y.-Y., N. Leporé, P. Saharan, S. K. Madsen, X. Hua, C. R. Jack, L. M. Shaw, J. Q. Trojanowski, M. W. Weiner, A. W. Toga *et al.*, "Ventricular maps in 804 ADNI subjects: correlations with CSF biomarkers and clinical decline", Neurobiology of aging **31**, 8, 1386–1400 (2010).

Chung, M. K., K. M. Dalton and R. J. Davidson, "Tensor-based cortical surface morphometry via weighted spherical harmonic representation", Medical Imaging, IEEE Transactions on **27**, 8, 1143–1151 (2008a).

Chung, M. K., K. M. Dalton and R. J. Davidson, "Tensor-based cortical surface morphometry via weighted spherical harmonic representation", IEEE Trans Med Imaging **27**, 8, 1143–1151 (2008b).

Chung, M. K., S. M. Robbins, K. M. Dalton, R. J. Davidson, A. L. Alexander and A. C. Evans, "Cortical thickness analysis in autism with heat kernel smoothing", NeuroImage **25**, 4, 1256–1265 (2005).

Chung, M. K., K. J. Worsley, T. Paus, C. Cherif, D. L. Collins, J. N. Giedd, J. L. Rapoport and A. C. Evans, "A unified statistical approach to deformation-based morphometry", Neuroimage **14**, 3, 595–606 (2001).

Chung, M. K., K. J. Worsley, S. Robbins, T. Paus, J. Taylor, J. N. Giedd, J. L. Rapoport and A. C. Evans, "Deformation-based surface morphometry applied to gray matter deformation", Neuroimage **18**, 2, 198–213 (2003).

Coates, A. and A. Y. Ng, "The importance of encoding versus training with sparse coding and vector quantization", in "Proceedings of the 28th International Conference on Machine Learning (ICML-11)", pp. 921–928 (2011).

Colliot, O., G. Chételat, M. Chupin, B. Desgranges, B. Magnin, H. Benali, B. Dubois, L. Garnero, F. Eustache and S. Lehéricy, "Discrimination between alzheimer disease, mild cognitive impairment, and normal aging by using automated segmentation of the hippocampus", Radiology **248**, 1, 194–201 (2008).

Combettes, P. L. and V. R. Wajs, "Signal Recovery by Proximal Forward-Backward Splitting", Multiscale Modeling & Simulation **4**, 4, 1168–1200 (2005a).

Combettes, P. L. and V. R. Wajs, "Signal recovery by proximal forward-backward splitting", Multiscale Modeling & Simulation **4**, 4, 1168–1200 (2005b).

Cuingnet, R., E. Gerardin, J. Tessieras, G. Auzias, S. Lehéricy, M.-O. Habert, M. Chupin, H. Benali and O. Colliot, "Automatic classification of patients with Alzheimer's disease from structural MRI: a comparison of ten methods using the adni database", neuroimage **56**, 2, 766–781 (2011).

Daubechies, I., M. Defrise and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint", Communications on Pure and Applied Mathematics **57**, 11, 1413–1457 (2004).

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database", in "Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on", pp. 248–255 (IEEE, 2009).

Donoho, D. L. and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization", Proc. Natl. Acad. Sci. U.S.A. **100**, 5, 2197–2202 (2003).

Evgeniou, T., C. A. Micchelli and M. Pontil, "Learning multiple tasks with kernel methods", in "Journal of Machine Learning Research", pp. 615–637 (2005).

Evgeniou, T. and M. Pontil, "Regularized multi–task learning", in "Proc. SIGKDD", pp. 109–117 (ACM, 2004).

Fan, Y., D. Shen and C. Davatzikos, "Classification of structural images via high-dimensional image warping, robust feature extraction, and SVM", Med Image Comput Comput Assist Interv **8**, Pt 1, 1–8 (2005).

Fan, Y., D. Shen, R. C. Gur, R. E. Gur and C. Davatzikos, "COMPARE: classification of morphological patterns using adaptive regional elements", IEEE Trans Med Imaging **26**, 1, 93–105 (2007).

Fernández, D. and M. V. Solodov, "Local convergence of exact and inexact augmented lagrangian methods under the second-order sufficient optimality condition", SIAM Journal on Optimization **22**, 2, 384–407 (2012).

Fernando, C., D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel and D. Wierstra, "Pathnet: Evolution channels gradient descent in super neural networks", arXiv preprint arXiv:1701.08734 (2017).

Ferrarini, L., W. M. Palm, H. Olofsen, R. van der Landen, M. A. van Buchem, J. H. Reiber and F. Admiraal-Behloul, "Ventricular shape biomarkers for Alzheimer's disease in clinical MR images", Magnetic resonance in medicine **59**, 2, 260–267 (2008a).

Ferrarini, L., W. M. Palm, H. Olofsen, R. van der Landen, M. A. van Buchem, J. H. Reiber and F. Admiraal-Behloul, "Ventricular shape biomarkers for Alzheimer's disease in clinical MR images", Magn Reson Med **59**, 2, 260–267 (2008b).

Fischl, B., "Freesurfer", Neuroimage **62**, 2, 774–781 (2012).

Folstein, M. F., S. E. Folstein and P. R. McHugh, ""mini-mental state": a practical method for grading the cognitive state of patients for the clinician", Journal of psychiatric research **12**, 3, 189–198 (1975).

Frisoni, G. B., N. C. Fox, C. R. Jack, P. Scheltens and P. M. Thompson, "The clinical use of structural MRI in Alzheimer disease", Nat Rev Neurol **6**, 2, 67–77 (2010a).

Frisoni, G. B., N. C. Fox, C. R. Jack Jr, P. Scheltens and P. M. Thompson, "The clinical use of structural mri in Alzheimer disease", Nature Reviews Neurology **6**, 2, 67 (2010b).

Gerardin, E., G. Chételat, M. Chupin, R. Cuingnet, B. Desgranges, H.-S. Kim, M. Niethammer, B. Dubois, S. Lehéricy, L. Garnero *et al.*, "Multidimensional classification of hippocampal shape features discriminates alzheimer's disease and mild cognitive impairment from normal aging", Neuroimage **47**, 4, 1476–1486 (2009).

Gong, P., J. Ye and C.-s. Zhang, "Multi-stage multi-task feature learning", in "Advances in neural information processing systems", pp. 1988–1996 (2012).

Gregor, K. and Y. LeCun, "Learning fast approximations of sparse coding", in "ICML", pp. 399–406 (Omnipress, 2010).

Han, X., C. Xu and J. L. Prince, "A Moving Grid Framework for Geometric Deformable Models", Int J Comput Vis **84**, 1, 63–79 (2009).

He, F. S., Y. Liu, A. G. Schwing and J. Peng, "Learning to play in a day: Faster deep reinforcement learning by optimality tightening", arXiv preprint arXiv:1611.01606 (2016a).

He, K., X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition", in "CVPR", pp. 770–778 (2016b).

Hinton, G., O. Vinyals and J. Dean, "Distilling the knowledge in a neural network", arXiv preprint arXiv:1503.02531 (2015).

Hinz, T., N. Navarro-Guerrero *et al.*, "Speeding up the hyperparameter optimization of deep convolutional neural networks", International Journal of Computational Intelligence and Applications p. 1850008 (2018).

Hitziger, S., M. Clerc, A. Gramfort, S. Saillet, C. Bénar and T. Papadopoulo, "Jitter-adaptive dictionary learning-application to multi-trial neuroelectric signals", arXiv preprint arXiv:1301.3611 (2013).

Hoerl, A. E. and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems", Technometrics **12**, 1, 55–67 (1970a).

Hoerl, A. E. and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems", Technometrics **12**, 1, 55–67 (1970b).

Huang, G., Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely connected convolutional networks.", in "CVPR", vol. 1-2, p. 3 (2017).

Huo, Y., A. J. Plassard, A. Carass, S. M. Resnick, D. L. Pham, J. L. Prince and B. A. Landman, "Consistent cortical reconstruction and multi-atlas brain segmentation", NeuroImage **138**, 197–210 (2016).

Ito, K., S. Ahadieh, B. Corrigan, J. French, T. Fullerton and T. Tensfeldt, "Disease progression meta-analysis model in Alzheimer's disease", Alzheimer's & Dementia **6**, 1, 39–53 (2010).

Jack Jr, C. R., M. A. Bernstein, N. C. Fox, P. Thompson, G. Alexander, D. Harvey, B. Borowski, P. J. Britson, J. L. Whitwell, C. Ward *et al.*, "The Alzheimer's Disease Neuroimaging Initiative (ADNI): MRI methods", Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine **27**, 4, 685–691 (2008).

Jiang, X., X. Li, J. Lv, T. Zhang, S. Zhang, L. Guo and T. Liu, "Sparse representation of HCP grayordinate data reveals novel functional architecture of cerebral cortex", Hum Brain Mapp **36**, 12, 5301–5319 (2015a).

Jiang, X., X. Li, J. Lv, T. Zhang, S. Zhang, L. Guo and T. Liu, "Sparse representation of hcp grayordinate data reveals novel functional architecture of cerebral cortex", Human brain mapping **36**, 12, 5301–5319 (2015b).

Jung, H., J. Ju, M. Jung and J. Kim, "Less-forgetful learning for domain expansion in deep neural networks", arXiv preprint arXiv:1711.05959 (2017).

Kirkpatrick, J., R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks", Proceedings of the national academy of sciences p. 201611835 (2017).

Klöppel, S., C. M. Stonnington, C. Chu, B. Draganski, R. I. Scahill, J. D. Rohrer, N. C. Fox, C. R. Jack Jr, J. Ashburner and R. S. Frackowiak, "Automatic classification of MR scans in Alzheimer's disease", Brain **131**, 3, 681–689 (2008).

Krizhevsky, A. and G. Hinton, "Learning multiple layers of features from tiny images", Tech. rep., Citeseer (2009).

Krizhevsky, A., I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in "Advances in neural information processing systems", pp. 1097–1105 (2012).

LeCun, Y., "The mnist database of handwritten digits", http://yann. lecun. com/exdb/mnist/ (1998).

Lee, H., A. Battle, R. Raina and A. Y. Ng, "Efficient sparse coding algorithms", in "Advances in neural information processing systems", pp. 801–808 (2006).

Li, Q., S. Qiu, S. Ji, P. M. Thompson, J. Ye and J. Wang, "Parallel lasso screening for big data optimization", in "Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", pp. 1705–1714 (ACM, 2016a).

Li, Q., T. Yang, L. Zhan, D. P. Hibar, N. Jahanshad, Y. Wang, J. Ye, P. M. Thompson and J. Wang, "Large-scale collaborative imaging genetics studies of risk genetic factors for alzheimer?s disease across multiple institutions", in "International Conference on Medical Image Computing and Computer-Assisted Intervention", pp. 335–343 (Springer, 2016b).

Li, X., X. Gu and H. Qin, "Surface mapping using consistent pants decomposition", IEEE Trans Vis Comput Graph **15**, 4, 558–571 (2009).

Li, Y., H. Chen, X. Jiang, X. Li, J. Lv, M. Li and et al., "Transcriptome Architecture of Adult Mouse Brain Revealed by Sparse Coding of Genome-Wide In Situ Hybridization Images", Neuroinformatics **15**, 3, 285–295 (2017).

Li, Z. and D. Hoiem, "Learning without forgetting", IEEE Transactions on Pattern Analysis and Machine Intelligence (2017).

Lin, B., Q. Li, Q. Sun, M.-J. Lai, I. Davidson, W. Fan and J. Ye, "Stochastic coordinate coding and its application for drosophila gene expression pattern annotation", arXiv preprint arXiv:1407.8147 (2014).

Lin, Z., M. Chen and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices", arXiv preprint arXiv:1009.5055 (2010).

Liu, J., S. Ji and J. Ye, "Multi-task feature learning via efficient l 2, 1-norm minimization", in "Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence", pp. 339–348 (AUAI Press, 2009a).

Liu, J., S. Ji and J. Ye, "SLEP: Sparse learning with efficient projections", Arizona State University URL `https://github.com/jiayuzhou/SLEP` (2009b).

Liu, M., D. Zhang and D. Shen, "Identifying informative imaging biomarkers via tree structured sparse learning for ad diagnosis", Neuroinformatics **12**, 3, 381–394 (2014).

Liu, X., M. Masana, L. Herranz, J. Van de Weijer, A. M. Lopez and A. D. Bagdanov, "Rotate your networks: Better weight consolidation and less catastrophic forgetting", arXiv preprint arXiv:1802.02950 (2018).

Liu, Y., T. Paajanen, Y. Zhang, E. Westman, L. O. Wahlund, A. Simmons, C. Tunnard, T. Sobow, P. Mecocci, M. Tsolaki, B. Vellas, S. Muehlboeck, A. Evans, C. Spenger, S. Lovestone and H. Soininen, "Combination analysis of neuropsychological tests and structural MRI measures in differentiating AD, MCI and control groups–the AddNeuroMed study", Neurobiol. Aging **32**, 7, 1198–1206 (2011).

Lopez-Paz, D. *et al.*, "Gradient episodic memory for continual learning", in "Advances in Neural Information Processing Systems", pp. 6467–6476 (2017).

Lorensen, W. E. and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm", in "ACM siggraph computer graphics", vol. 21-4, pp. 163–169 (ACM, 1987).

Lv, J., X. Jiang, X. Li, D. Zhu, H. Chen, T. Zhang, S. Zhang, X. Hu, J. Han, H. Huang, J. Zhang, L. Guo and T. Liu, "Sparse representation of whole-brain fMRI signals for identification of functional networks", Med Image Anal **20**, 1, 112–134 (2015a).

Lv, J., X. Jiang, X. Li, D. Zhu, S. Zhang, S. Zhao, H. Chen, T. Zhang, X. Hu, J. Han, J. Ye, L. Guo and T. Liu, "Holistic atlases of functional networks and interactions reveal reciprocal organizational architecture of cortical function", IEEE Trans Biomed Eng **62**, 4, 1120–1131 (2015b).

Lv, J., B. Lin, Q. Li, W. Zhang, Y. Zhao, X. Jiang, L. Guo, J. Han, X. Hu, C. Guo, J. Ye and T. Liu, "Task fMRI data analysis based on supervised stochastic coordinate coding", Med Image Anal **38**, 1–16 (2017).

Magnin, B., L. Mesrob, S. Kinkingnéhun, M. Pélégrini-Issac, O. Colliot, M. Sarazin, B. Dubois, S. Lehéricy and H. Benali, "Support vector machine-based classification of Alzheimers disease from whole-brain anatomical MRI", Neuroradiology **51**, 2, 73–83 (2009).

Mairal, J., F. Bach, J. Ponce and G. Sapiro, "Online dictionary learning for sparse coding", in "ICML", pp. 689–696 (2009).

Mairal, J., F. Bach, J. Ponce, G. Sapiro and A. Zisserman, "Discriminative learned dictionaries for local image analysis", in "Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit", pp. 1–8 (2008).

Meinshausen, N., P. Bühlmann *et al.*, "High-dimensional graphs and variable selection with the lasso", The annals of statistics **34**, 3, 1436–1462 (2006).

Moenning, C. and N. A. Dodgson, "Fast Marching farthest point sampling", Tech. Rep. UCAM-CL-TR-562, University of Cambridge, Computer Laboratory, URL `https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-562.pdf` (2003).

Parikh, N., S. Boyd *et al.*, "Proximal algorithms", Foundations and Trends in Optimization **1**, 3, 127–239 (2014).

Patel, J. R., T. R. Shah, V. P. Shingadiya and V. B. Patel, "Comparison between breadth first search and nearest neighbor algorithm for waveguide path planning", Int. J. Research and Scientific Innovation **2**, 19–21 (2015).

Pham, H., M. Y. Guan, B. Zoph, Q. V. Le and J. Dean, "Efficient neural architecture search via parameter sharing", arXiv preprint arXiv:1802.03268 (2018).

Rebuffi, S.-A., A. Kolesnikov, G. Sperl and C. H. Lampert, "icarl: Incremental classifier and representation learning", in "CVPR", (2017).

Rey, A., *Lexamen clinique en psychologie* (Presses Universitaires de France; Paris, 1964).

Rojas, R., "Adaboost and the super bowl of classifiers a tutorial introduction to adaptive boosting", Freie University, Berlin, Tech. Rep (2009).

Rosen, W. G., R. C. Mohs and K. L. Davis, "A new rating scale for Alzheimer's disease.", The American journal of psychiatry (1984).

Rusu, A. A., N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu and R. Hadsell, "Progressive neural networks", arXiv preprint arXiv:1606.04671 (2016).

Saadi, K., N. L. Talbot and G. C. Cawley, "Optimally regularised kernel Fisher discriminant classification", Neural Netw **20**, 7, 832–841 (2007).

Scardapane, S., D. Comminiello, A. Hussain and A. Uncini, "Group sparse regularization for deep neural networks", Neurocomputing **241**, 81–89 (2017).

Schuster, M. and K. K. Paliwal, "Bidirectional recurrent neural networks", IEEE Transactions on Signal Processing **45**, 11, 2673–2681 (1997).

Schwarz, J., J. Luketina, W. M. Czarnecki, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu and R. Hadsell, "Progress & compress: A scalable framework for continual learning", arXiv preprint arXiv:1805.06370 (2018).

Shen, K.-K., J. Fripp, F. Mériaudeau, G. Chételat, O. Salvado and P. Bourgeat, "Detecting global and local hippocampal shape changes in Alzheimer's disease using statistical shape models", Neuroimage **59**, 3, 2155–2166 (2012).

Shi, J., C. M. Stonnington, P. M. Thompson, K. Chen, B. Gutman, C. Reschke, L. C. Baxter, E. M. Reiman, R. J. Caselli and Y. Wang, "Studying ventricular abnormalities in mild cognitive impairment with hyperbolic Ricci flow and tensor-based morphometry", NeuroImage **104**, 1–20 (2015).

Shi, J., P. M. Thompson, B. Gutman, Y. Wang, A. D. N. Initiative *et al.*, "Surface fluid registration of conformal representation: Application to detect disease burden and genetic influence on hippocampus", NeuroImage **78**, 111–134 (2013).

Shi, J. and Y. Wang, "Hyperbolic Wasserstein distance for shape indexing", IEEE Trans Pattern Anal Mach Intell (2019).

Shi, J., W. Zhang, M. Tang, R. J. Caselli and Y. Wang, "Conformal invariants for multiply connected surfaces: Application to landmark curve-based brain morphometry analysis", Med Image Anal **35**, 517–529 (2017).

Simon, N., J. Friedman, T. Hastie and R. Tibshirani, "A sparse-group lasso", Journal of Computational and Graphical Statistics **22**, 2, 231–245 (2013).

Simonyan, K. and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", arXiv preprint arXiv:1409.1556 (2014).

Stonnington, C. M., C. Chu, S. Klöppel, C. R. Jack Jr, J. Ashburner and R. S. Frackowiak, "Predicting clinical scores from magnetic resonance scans in Alzheimer's Disease", Neuroimage **51**, 4, 1405–1413 (2010).

Suk, H.-I., S.-W. Lee, D. Shen, A. D. N. Initiative *et al.*, "Deep sparse multi-task learning for feature selection in alzheimers disease diagnosis", Brain Structure and Function **221**, 5, 2569–2587 (2016).

Suk, H.-I., S.-W. Lee, D. Shen *et al.*, "Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis", NeuroImage **101**, 569–582 (2014).

Sukkar, R., E. Katz, Y. Zhang, D. Raunig and B. T. Wyman, "Disease progression modeling using hidden markov models", in "EMBC", pp. 2845–2848 (IEEE, 2012).

Sun, D., T. G. van Erp, P. M. Thompson, C. E. Bearden, M. Daley, L. Kushan, M. E. Hardt, K. H. Nuechterlein, A. W. Toga and T. D. Cannon, "Elucidating a magnetic resonance imaging-based neuroanatomic biomarker for psychosis: classification analysis using probabilistic brain atlas and machine learning algorithms", Biol. Psychiatry **66**, 11, 1055–1060 (2009).

Sutton, R. S., D. A. McAllester, S. P. Singh and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation", in "Advances in neural information processing systems", pp. 1057–1063 (2000).

Thompson, P. M., J. N. Giedd, R. P. Woods, D. MacDonald, A. C. Evans and A. W. Toga, "Growth patterns in the developing brain detected by using continuum mechanical tensor maps", Nature **404**, 6774, 190–193 (2000).

Thompson, P. M., K. M. Hayashi, G. I. De Zubicaray, A. L. Janke, S. E. Rose, J. Semple, M. S. Hong, D. H. Herman, D. Gravano, D. M. Doddrell and A. W. Toga, "Mapping hippocampal and ventricular change in Alzheimer disease", Neuroimage **22**, 4, 1754–1766 (2004a).

Thompson, P. M., K. M. Hayashi, G. I. De Zubicaray, A. L. Janke, S. E. Rose, J. Semple, M. S. Hong, D. H. Herman, D. Gravano, D. M. Doddrell *et al.*, "Mapping hippocampal and ventricular change in alzheimer disease", Neuroimage **22**, 4, 1754–1766 (2004b).

Thompson, P. M., D. MacDonald, M. S. Mega, C. J. Holmes, A. C. Evans and A. W. Toga, "Detection and mapping of abnormal brain structure with a probabilistic atlas of cortical surfaces", J Comput Assist Tomogr **21**, 4, 567–581 (1997).

Tibshirani, R., "Regression shrinkage and selection via the LASSO", Journal of the Royal Statistical Society, Series B **58**, 267–288 (1994).

Tibshirani, R., "Regression shrinkage and selection via the lasso", Journal of the Royal Statistical Society. Series B (Methodological) pp. 267–288 (1996a).

Tibshirani, R., "Regression shrinkage and selection via the lasso", Journal of the Royal Statistical Society: Series B (Methodological) **58**, 1, 267–288 (1996b).

Tseng, P., "Convergence of a block coordinate descent method for nondifferentiable minimization", Journal of optimization theory and applications **109**, 3, 475–494 (2001).

Tsui, A., D. Fenton, P. Vuong, J. Hass, P. Koehl, N. Amenta, D. Coeurjolly, C. DeCarli and O. Carmichael, "Globally optimal cortical surface matching with exact landmark correspondence", Inf Process Med Imaging **23**, 487–498 (2013).

Van Essen, D. C., "Cortical cartography and caret software", Neuroimage **62**, 2, 757–764 (2012).

Vounou, M., T. E. Nichols, G. Montana, A. D. N. Initiative *et al.*, "Discovering genetic associations with high-dimensional neuroimaging phenotypes: A sparse reduced-rank regression approach", Neuroimage **53**, 3, 1147–1159 (2010).

Wah, C., S. Branson, P. Welinder, P. Perona and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset", Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011).

Wang, H., F. Nie, H. Huang, S. Risacher, C. Ding, A. J. Saykin and et al., "Sparse multi-task regression and feature selection to identify brain imaging predictors for memory performance", in "Proc IEEE Int Conf Comput Vis", pp. 557–562 (2011a).

Wang, L., J. S. Swank, I. E. Glick, M. H. Gado, M. I. Miller, J. C. Morris and J. G. Csernansky, "Changes in hippocampal volume and shape across time distinguish dementia of the Alzheimer type from healthy aging", Neuroimage **20**, 2, 667–682 (2003).

Wang, T., R. G. Qiu and M. Yu, "Predictive modeling of the progression of alzheimers disease with recurrent neural networks", Scientific reports **8** (2018).

Wang, Y., T. F. Chan, A. W. Toga and P. M. Thompson, "Multivariate tensor-based brain anatomical surface morphometry via holomorphic one-forms", in "International Conference on Medical Image Computing and Computer-Assisted Intervention", pp. 337–344 (Springer, 2009).

Wang, Y., L. Yuan, J. Shi, A. Greve, J. Ye, A. W. Toga, A. L. Reiss and P. M. Thompson, "Applying tensor-based morphometry to parametric surfaces can improve MRI-based disease diagnosis", Neuroimage **74**, 209–230 (2013).

Wang, Y., J. Zhang, B. Gutman, T. F. Chan, J. T. Becker, H. J. Aizenstein, O. L. Lopez, R. J. Tamburo, A. W. Toga and P. M. Thompson, "Multivariate tensor-based morphometry on surfaces: application to mapping ventricular abnormalities in HIV/AIDS", Neuroimage **49**, 3, 2141–2157 (2010).

Wang, Y. *et al.*, "Surface-based TBM boosts power to detect disease effects on the brain: an N=804 ADNI study", Neuroimage **56**, 4, 1993–2010 (2011b).

Weiner, M. W., D. P. Veitch, P. S. Aisen, L. A. Beckett, N. J. Cairns, R. C. Green and et al., "The Alzheimer's Disease Neuroimaging Initiative: a review of papers published since its inception", Alzheimers Dement **8**, 1 Suppl, 1–68 (2012).

Weiner, M. W., D. P. Veitch, P. S. Aisen, L. A. Beckett, N. J. Cairns, R. C. Green, D. Harvey, C. R. Jack, W. Jagust, E. Liu *et al.*, "The alzheimer's disease neuroimaging initiative: a review of papers published since its inception", Alzheimer's & Dementia **9**, 5, e111–e194 (2013).

Wu, J., J. Zhang, J. Shi, K. Chen, R. J. Caselli, E. M. Reiman and Y. Wang, "Hippocampus morphometry study on pathology-confirmed alzheimer's disease patients with surface multivariate morphometry statistics", in "2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)", pp. 1555–1559 (IEEE, 2018).

Wu, T. T. and K. Lange, "Coordinate Descent Algorithms for LASSO Penalized Regression", The Annals of Applied Statistics **2**, 1, 224–244 (2008).

Yang, J., J. Wright, T. S. Huang and Y. Ma, "Image super-resolution via sparse representation", IEEE Trans Image Process **19**, 11, 2861–2873 (2010).

Yin, W., S. Osher, D. Goldfarb and J. Darbon, "Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing", SIAM Journal on Imaging sciences **1**, 1, 143–168 (2008).

Yoon, J., E. Yang, J. Lee and S. J. Hwang, "Lifelong learning with dynamically expandable networks", arXiv preprint arXiv:1708.01547 (2017).

Yuan, L., Y. Wang, P. M. Thompson, V. A. Narayan and J. Ye, "Multi-source feature learning for joint analysis of incomplete multiple heterogeneous neuroimaging data", Neuroimage **61**, 3, 622–632 (2012).

Zeng, W., R. Shi, Y. Wang, S.-T. Yau, X. Gu and Alzheimer's Disease Neuroimaging Initiative, "Teichmüller shape descriptor and its application to alzheimer's disease study", Int. J. Comput. Vision **105**, 2, 155–170 (2013).

Zenke, F., B. Poole and S. Ganguli, "Continual learning through synaptic intelligence", arXiv preprint arXiv:1703.04200 (2017).

Zhang, D., D. Shen, A. D. N. Initiative *et al.*, "Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in alzheimer's disease", NeuroImage **59**, 2, 895–907 (2012).

Zhang, J., y. Fan, Q. Li, P. M. Thompson, J. Ye and Y. Wang, "Applying sparse coding to surface multivariate tensor-based morphometry to predict future cognitive decline", in "Biomedical Imaging (ISBI), 2017 IEEE 14th International Symposium on", (IEEE, 2017a).

Zhang, J., Y. Fan, Q. Li, P. M. Thompson, J. Ye and Y. Wang, "Empowering cortical thickness measures in clinical diagnosis of alzheimer's disease with spherical sparse coding", in "Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on", pp. 446–450 (IEEE, 2017b).

Zhang, J., Q. Li, R. J. Caselli, P. M. Thompson, J. Ye and Y. Wang, "Multi-source multi-target dictionary learning for prediction of cognitive decline", in "IPMI", pp. 184–197 (Springer, 2017c).

Zhang, J., Q. Li, R. J. Caselli, J. Ye and Y. Wang, "Multi-task dictionary learning based convolutional neural network for computer aided diagnosis with longitudinal images", arXiv preprint arXiv:1709.00042 (2017d).

Zhang, J., J. Shi, C. Stonnington, Q. Li, B. A. Gutman, K. Chen, E. M. Reiman, R. Caselli, P. M. Thompson, J. Ye *et al.*, "Hyperbolic space sparse coding with its application on prediction of alzheimer?s disease in mild cognitive impairment", in "International Conference on Medical Image Computing and Computer-Assisted Intervention", pp. 326–334 (Springer, 2016a).

Zhang, J., C. Stonnington, Q. Li, J. Shi, R. J. Bauer, B. A. Gutman, K. Chen, E. M. Reiman, P. M. Thompson, J. Ye and Y. Wang, "Applying sparse coding to surface multivariate tensor-based morphometry to predict future cognitive decline", Proc IEEE Int Symp Biomed Imaging **2016**, 646–650 (2016b).

Zhang, J., C. Stonnington, Q. Li, J. Shi, R. J. Bauer, B. A. Gutman, K. Chen, E. M. Reiman, P. M. Thompson, J. Ye *et al.*, "Applying sparse coding to surface multivariate Tensor-based morphometry to predict future cognitive decline", in "ISBI", pp. 646–650 (IEEE, 2016c).

Zhang, J., Y. Tu, Q. Li, R. J. Caselli, P. M. Thompson, J. Ye and Y. Wang, "Multi-task sparse screening for predicting future clinical scores using longitudinal cortical thickness measures", in "Proc IEEE Int Symp Biomed Imaging", pp. 1406–1410 (2018a).

Zhang, J. and Y. Wang, "Continually modeling Alzheimers disease progression via deep multi-order preserving weight consolidation", in "International Conference on Medical Image Computing and Computer-Assisted Intervention", (Springer, 2019a).

Zhang, J. and Y. Wang, "Continually modeling alzheimers disease progression via deep multi-order preserving weight consolidation", in "International Conference on Medical Image Computing and Computer-Assisted Intervention", pp. 850–859 (Springer, 2019b).

Zhang, J. and Y. Wang, "Temporally adaptive-dynamic sparse network for modeling disease progression", in "The 2020 International Symposium on Biomedical Imaging (ISBI)", (IEEE, 2020).

Zhang, J., J. Zhang, S. Ghosh, D. Li, S. Tasci, L. Heck, H. Zhang and C.-C. J. Kuo, "Class-incremental learning via deep model consolidation", in "Winter Conference on Applications of Computer Vision (WACV)", (IEEE, 2020).

Zhang, K., S. Zhe, C. Cheng, Z. Wei, Z. Chen, H. Chen, G. Jiang, Y. Qi and J. Ye, "Annealed sparsity via adaptive and dynamic shrinking", in "SIGKDD", pp. 1325–1334 (ACM, 2016d).

Zhang, T., "Solving large scale linear prediction problems using stochastic gradient descent algorithms", in "Proceedings of the twenty-first international conference on Machine learning", p. 116 (ACM, 2004).

Zhang, W., R. Li, T. Zeng, Q. Sun, S. Kumar, J. Ye and S. Ji, "Deep model based transfer and multi-task learning for biological image analysis", in "Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", pp. 1475–1484 (ACM, 2015).

Zhang, W., J. Lv, X. Li, D. Zhu, X. Jiang, S. Zhang, Y. Zhao, L. Guo, J. Ye, D. Hu and T. Liu, "Experimental comparisons of sparse dictionary learning and independent component analysis for brain network inference from fMRI data", IEEE Trans Biomed Eng (2018b).

Zhang, Z., Y. Xie, F. Xing, M. McGough and L. Yang, "Mdnet: A semantically and visually interpretable medical image diagnosis network", in "CVPR", pp. 6428–6436 (2017e).

Zhou, J., J. Liu, V. A. Narayan and J. Ye, "Modeling disease progression via fused sparse group lasso", in "Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 1095–1103 (ACM, 2012).

Zhou, J., J. Liu, V. A. Narayan and J. Ye, "Modeling disease progression via multi-task learning", Neuroimage **78**, 233–248 (2013).

Zhou, J. T., K. Di, J. Du, X. Peng, H. Yang, S. J. Pan, I. W. Tsang, Y. Liu, Z. Qin and R. S. M. Goh, "Sc2net: Sparse lstms for sparse coding", in "Thirty-Second AAAI Conference on Artificial Intelligence", (2018).

Zhu, D., Q. Li, B. C. Riedel, N. Jahanshad, D. P. Hibar, I. M. Veerh, H. Walterh, L. Schmaalc, D. J. Veltmanc, D. Grotegerdf *et al.*, "Large-scale classification of major depressive disorder via distributed lasso", in "Proc. of SPIE Vol", vol. 10160, pp. 101600Y–1 (2017).

Zoph, B. and Q. V. Le, "Neural architecture search with reinforcement learning", arXiv preprint arXiv:1611.01578 (2016).

APPENDIX A

COORDINATE DESCENT FOR SOLVING LASSO PROBLEM

Given a data point $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n) \in \mathbb{R}^{m \times N}$ and a dictionary $\mathbf{D} \in \mathbb{R}^{m \times t}$, the lasso problem is given as follows:

$$\min_{\mathbf{Z}} f(\mathbf{Z}) = \frac{1}{2}||\mathbf{DZ} - \mathbf{X}||_2^2 + \lambda||\mathbf{Z}||_1, \tag{A.1}$$

where $\mathbf{Z} = (\mathbf{z}_1, \cdots, \mathbf{z}_t) \in \mathbb{R}^{N \times t}$.

Suppose we freeze all columns of $\mathbf{Z}$ except the j-th column $\mathbf{z}_j$ in Eq. A.1. Let $\mathbf{d}_j$ denote the j-th column of $\mathbf{D}$ and $d_{i,j}$ is the element of i-th row and j-th column in $\mathbf{D}$. Therefore, we have

$$\begin{aligned} &\min_{\mathbf{z}_j} \frac{1}{2}(d_{i,j}\mathbf{z}_j - \mathbf{x}_i)^2 + \lambda|\mathbf{z}_j| \\ =&\min_{\mathbf{z}_j} \frac{1}{2}(\mathbf{z}_j^2 - 2b_j\mathbf{z}_j + {b_j}^2) + \lambda|\mathbf{z}_j| \\ =&\min_{\mathbf{z}_j} \frac{1}{2}(\mathbf{z}_j - b_j)^2 + \lambda|\mathbf{z}_j|, \end{aligned} \tag{A.2}$$

where $b_j = \sum_{i=1}^{m}(\mathbf{x}_i - \sum_{k \neq j} d_{ik}\mathbf{z}_k)d_{ij}$ and we use the condition that each column of D is unit norm. Then z_j has an optimal solution: $\mathbf{z}_j = h_\lambda(b_j)$, where h_λ is a soft thresholding shrinkage function so called the proximal operator of the ℓ_1 norm (Combettes and Wajs, 2005a). The definition of h_λ is as follows:

$$h_\lambda(v) = \begin{cases} v + \lambda, & v < -\lambda \\ 0, & -\lambda \leq v \leq \lambda \\ v - \lambda, & v < \lambda \end{cases}$$

Note that $b_j = \mathbf{d}_j^T\mathbf{x}_i - \mathbf{d}_j^T\mathbf{D}\mathbf{z}_j + (\mathbf{d}_j^T\mathbf{d}_j)\mathbf{z}_j = \mathbf{d}_j^T(\mathbf{x}_i - \mathbf{D}\mathbf{z}_j) + \mathbf{z}_j$. Therefore, the computational cost of updating the j-th coordinate $\mathbf{z}_j$ depends on computing the vector $\mathbf{x}_i - \mathbf{Dz}$ and the inner product $\mathbf{d}_j^T(\mathbf{x}_i - \mathbf{Dz})$.

APPENDIX B

PROOF OF PROPOSITION 3.7

proof. For convenience, let $\epsilon = 1/2$. Then it is easy to see that we have $\frac{1}{2}(\sqrt{k(k+1)}(\sqrt{k}+\sqrt{k+1})) \geq k\sqrt{k}$. It follows that

$$a_{k+1} \leq a_k + \frac{1}{k^{1+1/2}} \leq a_k + \frac{2}{\sqrt{k(k+1)}(\sqrt{k}+\sqrt{k+1})} = a_k + \frac{2}{\sqrt{k}} - \frac{2}{\sqrt{k+1}}. \tag{B.1}$$

Let $b_n = a_n + \frac{2}{\sqrt{k}}$. Thus, $0 \leq b_{k+1} \leq b_k$ and hence, $b_k, k \geq 1$ are convergent. Let us say $b_k \to a^*$, that is, $a_k + \frac{2}{\sqrt{k}} \to a^*$. It follows that $a_k \to a^*$. □