Latent Representation Learning for Understanding Relationships in Computer Vision and Neuroimaging

by

Seong Jae Hwang

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy
(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2019

Date of final oral examination: 10/04/2019

The dissertation is approved by the following members of the Oral Committee:

Vikas Singh, Professor, Biostatistics and Medical Informatics
Barbara B. Bendlin, Associate Professor, School of Medicine
Mohit Gupta, Assistant Professor, Computer Sciences
Xiaojin Zhu, Professor, Computer Sciences

To My Family

ACKNOWLEDGMENTS

I would like to use this as an opportunity to thank those who have helped me in the time of need and supported me throughout my PhD career for the past five years. First of all, I would like to thank my PhD advisor Prof. Vikas Singh who has mentored me. He had supported me in so many ways in order to let me focus solely on my research and not worry about other nuances. As an advisor, he had patiently guided me so I could become an independent researcher. In the later part of the PhD years, he was fully supportive of the decision that I had made and helped me to the best he could. I can humbly speak that Vikas is the person (other than my family) who has positively influenced me the most in my life. I am absolutely certain that no other advisors would have guided me to where I am now, and I could not have asked for a better outcome.

I would like to also mention the following mentors. My first research experience was off to a great start as a project assistant thanks to the continuous support of Dr. Houri Vorperian and valuable guidance by Prof. Moo Chung. Prof. Sterling Johnson's collaborative spirit was essentially the basis of my research interest in Alzheimer's disease. Prof. Barbara Bendlin is the most thoughtful and kind person I got to know in Madison that just talking to her keeps your spirits up. It was incredible see how her kind words motivated me from time to time. I got a chance to experience the best internship and summer when Dr. Joonseok Lee was my host at Google Research. Thanks to Prof. Mark Craven and Dr. Louise Pape, I had an invaluable interdisciplinary research experience as a part of the CIBM predoctoral traineeship.

I was able to survive the last five years in Madison thanks to the following friends. I would like to thank Hyunwoo Kim again for his generous response to my random email that I had sent him before I came to Madison, and his passion for research, which is second to none, still inspires

me. Won Hwa Kim was the best friend, colleague, and life mentor I met in Madison, and I cannot thank him enough for so many things that he had helped me out with. Without Vamsi Ithapu's extensive knowledge in neuroimaging, I would not have been able to complete my first paper and attend my first conference. Nearly every project I have worked on has traces of Sathya Ravi's mathematical brilliance, and I believe many would feel the same for their projects. Whenever I wanted to have a normal (i.e., non-research) conversation with a sensible person, Ronak was the person to find. He was also the most dependable, trustworthy person in the lab whom I could talk to about anything. Thanks to Yunyang Xiong positive atmosphere, we could endure our depressing submission periods I would like to thank Zirui Tao for diligently working with me and lending me his amazing skills for our proud papers. Thanks to him, my expectations from undergraduate students are now quite high.

I would like to thank my partner Hyo Kyung Lee who has essentially turned my PhD years into the best five years of my life. I am lucky that we were and will be able to share every part of our unique experiences with each other. Lastly and most importantly, I would like to thank my family and Seong Do for continuously showing their support through many years of uncertainty.

The projects in the thesis are supported in part by NIH (R01AG040396, R01AG021155, R01AG027161, P50AG033514, R01AG059312, R01EB022883, R01AG062336), the Center for Predictive and Computational Phenotyping (U54AI117924), NSF CAREER Award (1252725), and the UW Computation and Informatics in Biology and Medicine fellowship (5T15LM007359-14).

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The importance of data in modern society is enormous and underlies most aspects of our daily lives, including technology, health, finance, and economy. But data in its raw form does not provide actionable knowledge and needs to be turned into something more usable. Specifically, these processes of refining data often derive *latent representations*. For example, one may use transformation procedures to extract hidden factors from data through shallow models such as component analysis and deep models such as deep neural networks. While the direct use of factor analysis methods continues to be useful, many modern problems in computer vision and neuroimaging rely on higher-level latent representations, better informed by the needs of the downstream analysis tasks, that may otherwise be difficult. For instance, we may seek to understand an image as humans do by learning the visual relationship between objects in images. To better understand the structural integrity of the brain suffering from a neurodegenerative disease, we may analyze the relationship between the brain regions. Deriving such higher-level latent representations requires methods that are capable of addressing diverse data- and domain-specific challenges in the following various aspects: (1) structure of data (e.g., graph, sequence, image), (2) relationship type (e.g., temporal visual, inter-modality), (3) problem type (e.g., classification, generation), and (4) domain-specific challenges (e.g., small sample size, skewed distribution). In this thesis, we demonstrate how various statistical and machine learning models of shallow and deep formulations can help us to better understand various relationships for computer vision and neuroimaging problems. For each problem, we propose a novel approach that takes advantage of both the traditional statistical properties and recently developed deep models, and show its effectiveness quantitatively and qualitatively for enabling robust learning tasks and scientific discoveries.

1 INTRODUCTION

"Data is the new oil" – while some call this a ludicrous proposition, a growing number of technology evangelists see strong parallels between the two: just like how oil powers numerous industrial systems of our society from transportation to agriculture, data is a new valuable resource with massive information warehouses enabling all kinds of data-driven systems that benefit our society. This analogy between data and oil was first publicly recognized in 2006 by a British data scientist named Clive Humby who had earlier pioneered the UK's supermarket loyalty card scheme in 1994. His customer data science company *dunnhumby* analyzed years of customer information within only a few months and formulated a scheme that eventually became the first successful supermarket loyalty card, Tesco Clubcard, and transformed the marketing landscape of supermarkets in the UK at the time. Today, it is not surprising to see how almost every company is acquiring data from customers to optimize important decision making processes. Often, some collect data at a large scale and operate as data trading companies, sometimes referred to as "data brokers", by mining data from customers and selling them to those who would analyze and make use of it for monetary gain. In many ways, data is becoming a commodity on its own for various data-driven applications, just like how oil is traded as a versatile resource for diverse end-products.

While the above examples demonstrate the power of how data can be used to significantly enable efficiency gains in various applications, they also underscore a crucial characteristic of data which also resembles that of oil: data cannot readily be used "as it is" in its crude, original representation. Instead, to maximize its value, it needs to be turned into something more usable, similar to how crude oil needs to be changed into gas, plastic, chemicals, etc. for different end-products. For instance, the underlying signal in data is almost always contaminated with noise, so we need to

distill it into its most important part. Also, data often pertains to multiple sources of information, so we sometimes need to *separate* the sources into different components (e.g., independent component analysis) similar to refining oil into liquids and gases. For high-dimensional data, we may want to *extract* the features that are more important than the others to summarize the data to its essence (e.g., principal component analysis). These processes of refining data into more informative and "usable" representations often involve a transformation procedure to extract latent factors from data to derive its latent representation.

Now, to better understand how raw data can become more "usable", we discuss a few examples of extracting latent representations from the data via various transformation methods. Consider a set of examples $\mathbf{X} \in \mathbb{R}^{n \times k}$ with n number of samples (rows) with k-dimensional features. For a large k, it is often difficult to intuitively understand the underlying pattern of the data. To facilitate this, Principal Component Analysis (PCA) (Pearson, 1901) finds an orthogonal basis $\mathbf{W} \in \mathbb{R}^{k \times m}$ called principal axes such that the projected *principal components* (PCs) $\mathbf{P} = \mathbf{X}\mathbf{W}^\top$ have a high variance in each component (column). Since m PCs with the highest variances are m left singular vectors of $\mathbf{X}$ with highest singular values, $\mathbf{X}$ with highly correlated variables is accurately characterized with only a few $m < k$ number of PCs (i.e., $\mathbf{X} \approx \mathbf{P}\mathbf{W}^\top$ for $\mathbf{P} \in \mathbb{R}^{n \times m}$). Thus, PCA performs a linear orthogonal transformation on $\mathbf{X}$ to derive $\mathbf{P}$, a latent representation purposed for disentangling correlated variables:

$$\mathbf{X} \xrightarrow{\text{PCA}} \mathbf{P} \quad \text{such that } \mathbf{P} \text{ has high variance features.}$$

Due to its simple and nonparametric nature, PCA has been widely utilized in various applications across domains (Abdi and Williams, 2010).

A related technique is Canonical Correlation Analysis (CCA) (Hotelling, 1936). Given two datasets $\mathbf{X} \in \mathbb{R}^{n \times k}$ and $\mathbf{Y} \in \mathbb{R}^{n \times l}$ of n identical samples (rows) with k and l features respectively, CCA finds vectors $\mathbf{a_1}$ and $\mathbf{b}_1$ that

maximize the correlation between $\mathbf{Xa_1}$ and $\mathbf{Yb}_1$. $\mathbf{U}_1 = \mathbf{Xa_1}$ and $\mathbf{V}_1 = \mathbf{Yb}_1$ are called the first pair of canonical variables, and the subsequent pairs are found recursively. Thus, CCA performs a series of linear transformations on $\mathbf{X}$ and $\mathbf{Y}$ to derive latent representations that capture the relationship (i.e., correlation) between the variables/features:

$$\mathbf{X}, \mathbf{Y} \xrightarrow{\text{CCA}} \mathbf{U}, \mathbf{V} \quad \text{such that } \mathbf{U} \text{ and } \mathbf{V} \text{ have highly correlated features.}$$

CCA has been applied to a range of applications with multiple data modalities (Hardoon et al., 2004).

Lastly, when data comes with labels, Linear Discriminant Analysis (LDA) (Fisher, 1936) is suitable which finds the principal axes that maximize the variance of the data *and* the separation between the classes. Technically close to PCA, LDA performs a linear orthogonal transformation to derive the latent representations (i.e., eigenvectors) that maximize the between-class variance and minimizes the within-class variance:

$$\mathbf{X} \xrightarrow{\text{LDA}} \mathbf{P} \quad \text{such that high variance } \mathbf{P} \text{ also separates classes.}$$

LDA has been a popular method for supervised pattern recognition problems (McLachlan, 2004).

These methods impose varying properties on the latent representations, but they are all constructed under a common premise: the representations are "latent" in a sense that the variables meaningfully encode the underlying structure or pattern of data (e.g., a correlation among the features). Thus, even though a latent representation itself is not necessarily "interpretable" (e.g., PCs are not directly explainable in terms of features), data in its latent representation is almost always more useful than its original form for any subsequent applications or analyses. In fact, this overall idea of latent representation learning has been pursued for years with these classical representation learning methods. These have been the

strong foundation of many data-driven applications and analyses since mid 1900s: data visualization (Gabriel, 1971), clustering (Bartlett, 1963), image understanding (Zhao et al., 1999), anomaly detection (Boráros and Boráros, 1969), image denoising (Hoštalkova and Procházka, 1905), and many more.

Specifically, these methods excel in cases when their assumptions on the data are appropriate. For instance, PCA and LDA assume linearly correlated features with minimal outliers. The correlation coefficient computed in CCA assumes a linear relationship between any two variables. In some cases, the assumptions are reasonable enough to carry out the analyses with these methods. However, in other cases that we will describe next, we need richer representations beyond the latent representations described above.

Over the last decade, a family of methods in Deep Learning (DL) has demonstrated impressive performance in deriving rich representations from complex data. Without imposing specific assumptions on the data, DL models are able to extract the underlying structure of the data through carefully structured neural networks that often can capture richer statistical properties/patterns in the data. For instance, to derive a latent representation $\mathbf{Z}$ while accounting for the *nonlinear* correlation among the features, an autoencoder (AE) (Hinton and Salakhutdinov, 2006) with neural networks f (encoder) and g (decoder) achieves the following:

$$\mathbf{X} \xrightarrow{\text{AE}} \mathbf{Z} \quad \text{such that } \mathbf{X} \xrightarrow{f} \mathbf{Z} \xrightarrow{g} \mathbf{X}' \text{ for } \mathbf{X}' \approx \mathbf{X}.$$

Here, $\mathbf{Z}$ is a latent representation of $\mathbf{X}$, and $\mathbf{Z}$ encodes the underlying structure of $\mathbf{X}$ such that $\mathbf{Z}$ can be decoded (reconstructed) back to $\mathbf{X}' \approx \mathbf{X}$. Typically, when f and g are simple single layer fully-connected neural networks with a linear activation function (i.e., $\mathbf{Z} = f(\mathbf{X}) = W\mathbf{X}$, $\mathbf{X}' = g(\mathbf{Z}) = V\mathbf{Z}$), the subspace spanned by an AE trained via a squared error loss becomes nearly identical to the one derived from PCA with the linearity assumption.

When the data has some nonlinear feature relationships (e.g., the lattice structure of pixel intensity features in an image (Krizhevsky and Hinton, 2011)), an AE can be extended to learn the underlying complex structure of the features via f and g performing nonlinear transformations with multilayer neural networks with nonlinear activation functions.

For data modalities with a unique structure specific to the data, deep neural networks can be appropriately formulated to extract such structures. This was successfully demonstrated for applications in Computer Vision where the overarching goal is to understand the images as humans do. For those applications, the spatial information that needs to be extracted from an image (e.g., shapes, locations, and orientation of an object) requires a method which exploits the geometric structure of the data (e.g., nearby pixels) and understands their relationship (e.g., how they construct certain shapes). A class of deep neural networks called convolutional neural network (CNN) is a special type of deep neural network which explicitly aims to derive a latent representation encoding such spatial dependencies in an image:

$$\mathbf{X} \xrightarrow{\text{CNN}} \mathbf{Z} \quad \text{such that } \mathbf{Z} \text{ encodes the spetially derived features } \mathbf{X}.$$

A precisely derived $\mathbf{Z}$ can then be used for various image understanding applications such as object detection (Ren et al., 2015), image classification (Krizhevsky et al., 2012), and face recognition (Lawrence et al., 1997).

Analogous to the spatial features of image data, the temporal dependency of sequential features is another type of crucial information found in sequential data such as videos. For instance, let $\mathbf{X}_1, \ldots, \mathbf{X}_T$ be a sequence of data of length T. An informative temporal latent representation $\mathbf{Z}_t$ at the current time point t would encode the information from the past ($\mathbf{X}_{<t}$) which a model utilizes for the future time points $> t$. A common variant of neural network which allows this formulation is called a recurrent neural network (RNN) (Cho et al., 2014) which recursively uses the current

information $\mathbf{X}_t$ and the past latent information $\mathbf{Z}_{t-1}$ to derive the newly updated $\mathbf{Z}_t$:

$$\mathbf{X}_t, \mathbf{Z}_{t-1} \xrightarrow{\text{RNN}} \mathbf{Z}_t \quad \text{such that } \mathbf{Z}_t \text{ encodes the temporal information.}$$

Variants of RNN such as Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Units (Chung et al., 2014) have been developed to better capture the complex temporal information of lengthy sequences for modern applications including language modeling and video understanding.

As demonstrated thus far, data is refined into numerous latent representations through shallow and deep models, powering various applications across domains. Yet, our focus has been on deriving latent representations with a high emphasis on the data and less emphasis on the application. But nowadays, modern applications aim to achieve more complex tasks by understanding the "higher-level" concepts of *relationships* from data which are much more difficult to capture and less trivial to numerically represent. For instance, consider a visual question answering problem (VQA) where given an image and a natural language question about the image, the task is to provide an accurate natural language answer (e.g., Q: What is the person holding? A: Books). For a complex problem like this, formally setting up the problem with appropriate datasets (e.g., a list of visual questions and answers) and problem formulations (e.g., multi-class classification of answers) is a crucial first step.

Then, we ask the following: How do we exactly represent the concept of 'holding' when 'person' and 'books' are visually interacting in such a way? In this case, an ideal latent representation would somehow be able to inform that 'holding' is the most sensible interaction between 'person' and 'books' in an image. Generally speaking, we want to do the following:

$$\mathbf{X} \rightarrow \mathbf{Z} \quad \text{such that } \mathbf{Z} \text{ encodes relationships of interest.}$$

Deriving such advanced latent representation is a delicate process which requires us to consider various aspects:

1. *Structure of data* - e.g., Is it a graph, sequence, or image?
2. *Relationship type* - e.g., Is it a temporal, visual, or intra/inter modality relationship?
3. *Problem type* - e.g., Is it a classification, generation, or statistical analysis problem?
4. *Domain-specific challenges* - e.g., Small sample in medical imaging, highly skewed distributions, multi-modal analysis.

Thus, constructing novel methods that derive informative latent representations of various types of relationships while addressing various challenges posed above is the key to accomplishing modern applications and data analysis. The effort to contribute to this challenging task is the main focus of this thesis. In the next section, we will go over various types of relationships that are found across diverse problems and see what they can enable if their latent representations are properly derived.

1.1 Capturing Relationships in Latent Space

Relationships, by nature, may often be neither evident nor directly observable through raw data as such associations are not always explicitly measured in its raw representation. For instance, many relationships that we know are based on deductive reasoning (e.g., wet ground suggests a recent shower) or statistical hypothesis testing (e.g., statistically significant association between blood pressure and cardiovascular diseases). While traditional tasks such as object detection can explicitly be demonstrated on an image (e.g., we can locate a glass in the image directly in Fig. 1.1),

constructing higher-level descriptions about the objects such as "*the glass is on the table*" in Fig. 1.1 is not a straightforward task.

We take computer vision as an example with Fig. 1.1. Modern computer vision tasks now aim to *understand* images as humans do by detecting **visual relationships between objects** through various types of latent representations with both shallow (tensor factorization (Hwang et al., 2018)) and deep (visual relationship detection network (Lu et al., 2016; Zhan et al., 2019)) models. These approaches derive both the latent representations of the objects and their relationships which allow them to find the most likely relationships given objects in an image.

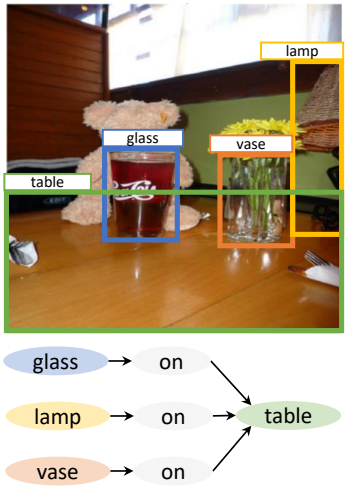
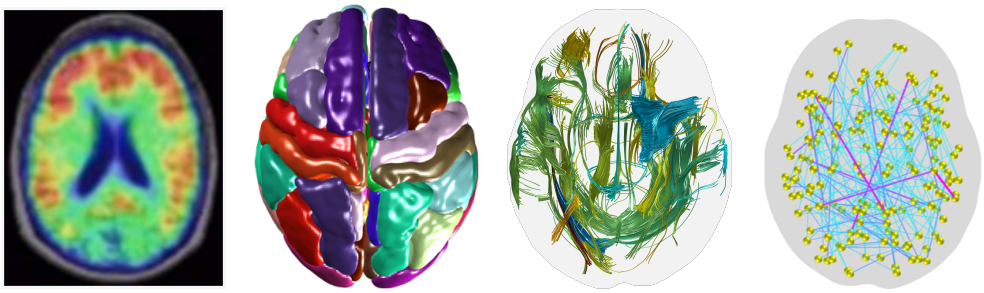


Figure 1.1: Detecting visual relationships between objects.

To understand the data as a whole, we need to take a step back and consider the **relationships between the samples or instances** of the data which provide useful knowledge and information about the data from different perspectives. For instance, the relationships between the samples can be constructed as a graph where the nodes are the samples and the edges represent the similarities between the pairs of samples. In such cases, the latent representation of the samples (e.g., the final layer output of a deep network (Cao et al., 2016; Monti et al., 2017)) can provide concise information that ordinary distance measures can be applied to them directly to construct graphs that better capture the relationships.

Exploiting the *graph structure* with its latent representation has also shown to be extremely beneficial for analyzing brain connectivity network (Fig. 1.2d) in neuroimaging (Fischer et al., 2015; Ma et al., 2017). Specifically, the structural integrity of a subject's brain can be inferred from the subject's brain connectivity network encoding the **relationship between**



(a) Amyloid PET (b) Desikan Atlas (c) Fiber Bundles (d) Brain Network

Figure 1.2: Illustrations of various brain imaging modalities. (a) Amyloid PET scan measure the level of amyloid beta accumulation in the brain. (b) Desikan atlas is one of the widely used gray matter regions predefined by their functions. (c) Fiber bundles are the neuron pathways within the white matter region structurally connecting various gray matter regions. (d) Brain network as a form of graph can be derived from the fiber bundles to measure the strengths of their structural connections (edges) between the regions (nodes).

brain regions. Although this neuroimaging modality is very informative in understanding the disease progression of neurodegenerative diseases such as Alzheimer's disease (Kim et al., 2015; Hwang et al., 2019a), the data often comes with several nuances. First, a brain imaging data almost always comes with an inevitable noise which is introduced during the data acquisition and preprocessing steps. Second, analyzing the data of cognitively healthy subjects is of great interest for an early understanding of the disease, but the disease progression at such early stages is very subtle and difficult to detect from the data. These analyses require a model which is sensitive and robust, and a unique latent representation on a graph called the *wavelet connectivity signature* has shown to reveal subtle characteristics of the brain network (Kim et al., 2015; Hwang et al., 2019a).

Further, understanding the **relationships between neuroimaging modalities** (shown in Fig. 1.2) is a crucial part of neuroscientific analyses to better understand the pathological process of neurodegenerative diseases (Racine et al., 2014; Guye et al., 2010). However, such analyses involving multiple modalities often need to explicitly account for the distinct characteristics of those modalities. For instance, MRI and PET scans, two of

the most common imaging modalities, have different spatial resolutions, so comparing and analyzing these modalities require a model which can explicitly account for the difference in feature dimensions. Fortunately, searching for the relationship between neuroimaging modalities with respect to their latent representations *directly* which more accurately encode the central pattern of the modalities may lead to novel associations that were often too subtle in the raw representation space (Hwang et al., 2019a; Kim et al., 2018).

The types of relationships described thus far do *not* explicitly assume a consistency or similarity between the samples. For instance, we do not explicitly (i.e., methodologically or by the model design) enforce the latent representations of two samples to be similar. However, for *sequential samples*, there is a natural constraint that we need to impose, which is the temporal or sequential consistency throughout the sequence (e.g., the representations of two consecutive time points should be consistent to a certain degree). By constructing the latent representations which capture the **temporal relationships between time points** of the sequential samples, the core temporal information can be preserved while still allowing the observed, natural variation throughout the sequence. For example, recurrent-type neural networks (Cho et al., 2014; Chung et al., 2014) and transformers (Devlin et al., 2019) can sequentially modulate the hidden information which passes through time points. Such latent representations can be especially useful in a *longitudinal neuroimaging analysis* where each subject has multiple time points of brain images which may show a potential underlying progression of the brain images (along with the noise variations) across the time points (Fig. 1.3). In such a case, a robust latent representation such as the harmonic bases can be directly used to model the sequential pattern which still captures the overall consistent patterns in certain brain regions while filtering out the inconsistent local patterns in other brain regions (Hwang et al., 2016).

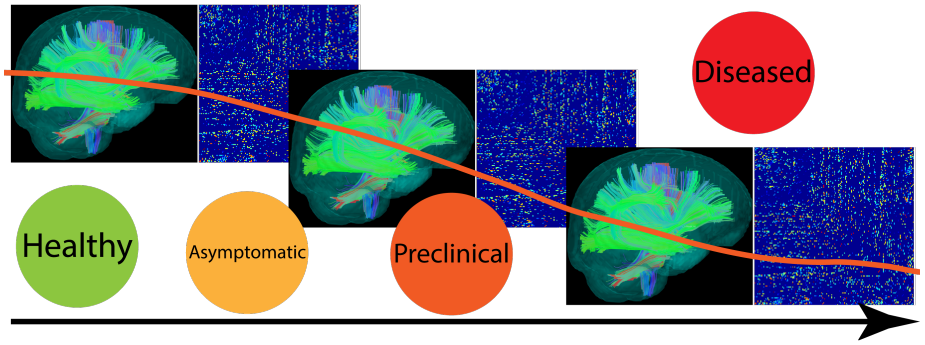


Figure 1.3: Longitudinal progression of brain network in Alzheimer's disease.

Lastly, we point out a unique problem where we need to learn multiple types of relationships and multiple types of latent representations. First, our samples are sequences which require **temporal relationships** through temporal latent representations. Second, we look for the **relationships between two sequential modalities** such as brain image sequences and cognitive function sequences through a latent mapping involving latent representations. Finally, we construct a bijective mapping of **relationship between the sample and the latent space**. Interestingly, a model which holistically captures all these types of relationships can *generate sequential samples given sequential conditions*, and it has a wide range of applications especially in neuroimaging. For instance (Fig. 1.8), given a series of cognitive scores (sequential condition), we may generate random brain image sequence samples that realistically follow the pattern of the brain image sequences (sequential samples) of those with similar cognitive scores (Hwang et al., 2019c). This unique formulation is an excellent example of how latent representations can be cleverly utilized to not only model *cross-modal and temporal relationships* but also explicitly use the *latent representation as a part of the relationship* that we model to enable a challenging task of conditional sequence generations.

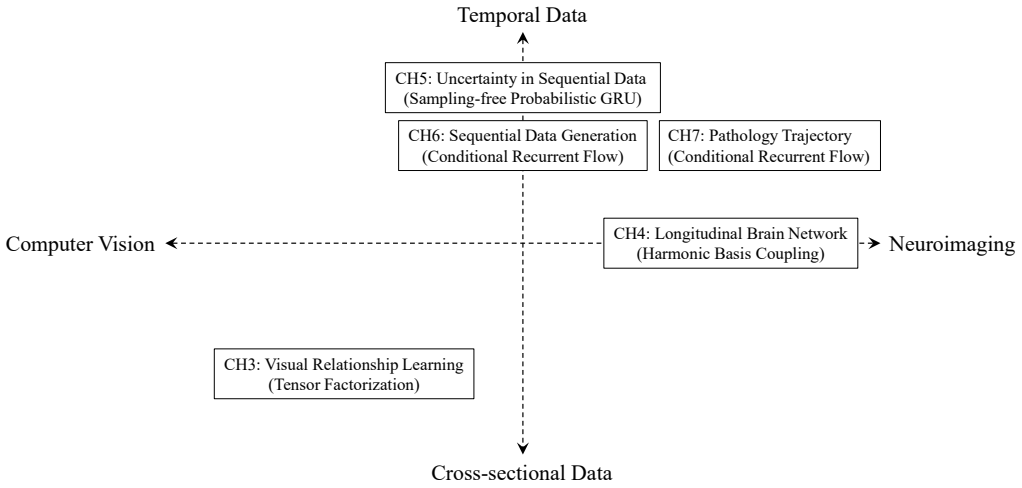


Figure 1.4: Overall scope of the dissertation

1.2 Contributions and Thesis Structure

The list spans a wide range of distinct domains with a common goal of understanding and modeling various domain-specific relationships with latent representation learning. These "high-level" tasks, if accomplished, may lead to much more impactful outcomes for their domain (e.g., identifying brain regions associated with early AD progression) beyond what was possible with the original data directly. Despite their common aims, it is challenging to simply use an existing family of models, including deep learning architectures, in an off-the-shelf manner appropriately. Thus, the contribution of this dissertation is as follows: *to develop statistical and machine learning models that can learn latent representations and understand various relationships while effectively addressing data- and domain-specific challenges.* In Fig. 1.4, we show the overall scope of the dissertation along the *domain* axis from computer vision to neuroimaging and the *data* axis from cross-sectional to temporal modalities. From the list of applications above in computer vision and neuroimaging which seek to understand and discover the relationships of, we now provide a brief background to the

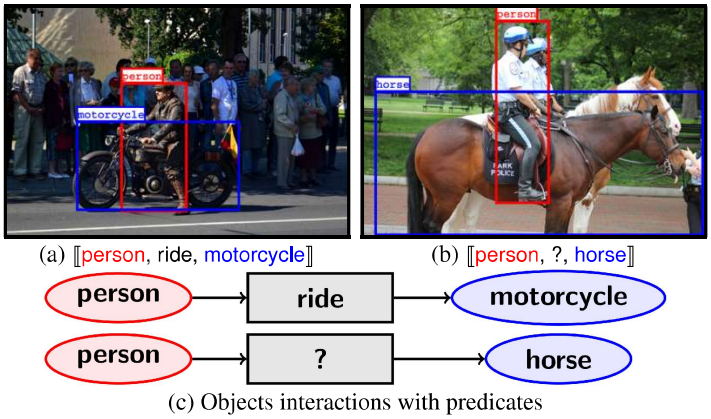


Figure 1.5: Main idea of Chapter 3: Visual relationship detection. The goal is to infer visual relationships that best describe the interactions among those objects. (a): A relationship instance in a training set. (b): An unknown relationship to predict. (c): The interactions of the objects (i.e., motorcycle and horse are both 'ridable') can be used to infer the correct relationship.

specific problems that we tackle, identify their unique challenges, and introduce how we will tackle them later in each chapter.

CH 3. Capturing the Latent Representations of Visual Relationships in Computer Vision

Given a set of localized objects in some training data, visual relationship detection seeks to detect the most likely "relationship" between objects in a given image (among all possible object pairs and their relations, Fig. 1.5). For instance, Fig. 1.1 shows an example of a *scene graph* of the given image which is a graph of visual relationship between the objects in the given image (e.g., [[*glass, on, table*]]). While the specific objects may be well represented in the training data, their relationships may still be infrequent. The empirical distribution obtained from seeing these relationships in a dataset does not model the underlying distribution well which is a serious issue for most learning methods (both shallow and deep).

In other words, the **challenge** is that despite the combinatorially many

possible relationship (e.g., N possible object categories and M possible predicate leads to N^2M number of [[*object,predicate,object*]] relationship tuples), not all possible relationships are observed even in the largest visual relationship data available (Krishna et al., 2016). Thus, a method which can successfully estimate the under-represented (i.e., less observed in the data) or unobserved relationships is essential for robustly learning the combinatorially large space of visual relationships. In **Chapter 3**, we will describe how our multi-relational learning model using a novel tensor factorization approach on a tensor representation of the relationship data can derive the *latent representations of both the objects and predicates* and provide both empirical improvements and theoretical guarantees (Hwang et al., 2018). We will also show how such "shallow" model can effectively regularize a deep model (i.e., deep neural networks) to take advantage of both sides.

CH 4. Coupling Harmonic Representations of Graphs to Characterize Cross-sectional and Longitudinal Relationships of Brain Connectivity

There is a great deal of interest in using large scale brain imaging studies to understand how brain connectivity evolves for an individual and how it varies over different levels/quantiles of cognitive function. To do so, one typically performs so-called tractography procedures on diffusion MR brain images (Fig. 1.2c) and derives measures of brain connectivity expressed as graphs (Fig. 1.2d). The nodes correspond to distinct brain regions (Fig. 1.2b) and the edges encode the strength of the connection (Fig. 1.2c). The scientific interest is in characterizing the evolution of these graphs (1) of the individuals over time (temporal progression within subjects) *and* (2) of the group from healthy to diseased (cross-sectional progression across the subject group). Fig. 1.6 illustrates such

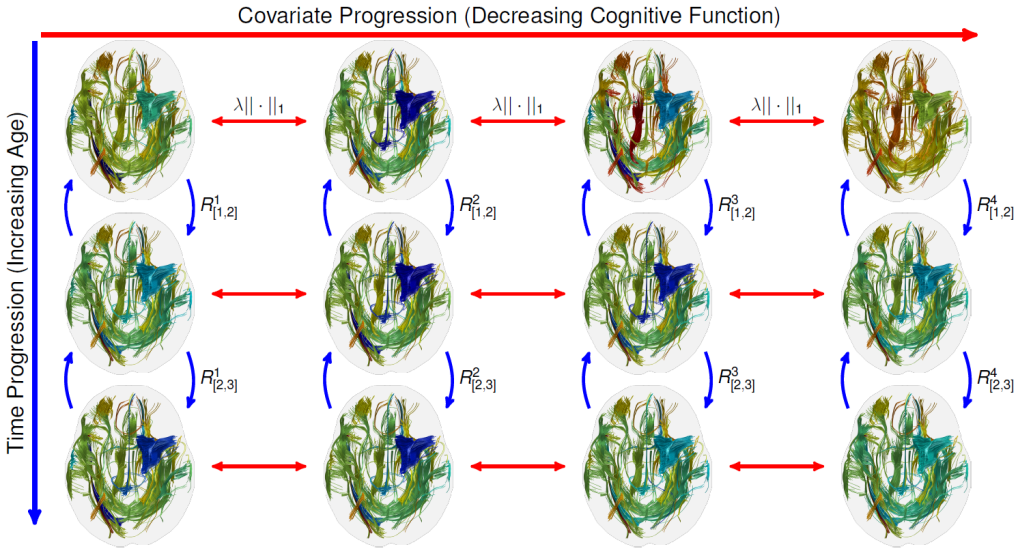


Figure 1.6: Main idea of Chapter 4: Characterizing cross-sectional and longitudinal progression of brain connectivity. Here, we show the evolution of top 50 most changing fiber tracts of the real data derived from the coupled harmonic bases. The tract colors represent their strong (blue) and weak (red) connectivity strengths. Cross-sectional coupling (red arrows) via ℓ_1-norm in each row. Longitudinal coupling (blue arrows) via rotation constraints in each column.

cross-sectional and temporal progression of brain connectivity we seek to characterize.

We are specifically interested in understanding the connectivity pattern of early stages of Alzheimer's disease (AD) of cognitively healthy subjects, or preclinical subjects, who are at risk of developing AD. However, the **challenge** comes from the subtle biomarker abnormality (including brain connectivity) at the early stage which makes them extremely difficult to characterize based on the raw measurements. In **Chapter 4**, we will describe how we draw cross-sectional and temporal associations from the subtle connectivity signals of a preclinical AD cohort in their *harmonic bases representations* via bases coupling (Hwang et al., 2015, 2016).

CH 5. Estimating Uncertainty in Latent Representations for Longitudinal Predictions

Although characterizing the longitudinal progression of brain network was described in the previous chapter, making sensible predictions *beyond* the observed time points (i.e., future predictions) may require more sophisticated models that specialize in sequential predictions. Specifically, an ideal model should be capable of effectively utilizing the temporal information from the past to make accurate future predictions. Recently, a family of sequential neural networks called recurrent neural networks (RNN) and its variants have shown promising results on such sequential prediction tasks using long and high-dimensional sequential data (Chung et al., 2014).

But what is more important in practice is that without acknowledging the level of uncertainty about the prediction, the model cannot be entirely trusted in scientific applications. For instance, unexpected performance variations with no sensible way of anticipating this possibility may also be a limitation in terms of regulatory compliance. When a decision made by a model could result in dangerous outcomes in real-life tasks such as an autonomous vehicle not detecting a pedestrian (Aguiar and Hespanha, 2007) or missing a disease prediction due to some artifacts in a medical image (Leibig et al., 2017; Nair et al., 2018), knowing how 'certain' the model is about its decision can offer a chance to look for alternative solutions such as alerting the driver to take over or recommending a different disease test to prevent undesirable outcomes made by erroneous decisions.

In fact, there has recently been a concerted effort to derive mechanisms in vision and machine learning systems to offer uncertainty estimates of the predictions they make (Gal and Ghahramani, 2016; Fortunato et al., 2017). Clearly, there are benefits to a system that is not only accurate but also has a sense for when it is not. Existing proposals center around Bayesian interpretations of modern deep architectures – these are effective but can often

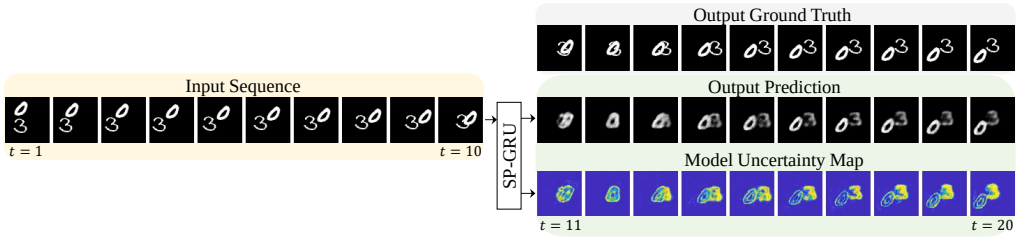


Figure 1.7: Image sequence prediction with uncertainty. Given the first 10 frames of an input sequence (left), our model SP-GRU makes the Output Prediction and the pixel-level Model Uncertainty Map where bright regions indicate high uncertainty. SP-GRU estimates the uncertainty *deterministically* without sampling model parameters.

be computationally demanding. In **Chapter 5**, we show how classical ideas in the literature on *exponential families* on probabilistic networks provide an excellent starting point to derive uncertainty estimates in Gated Recurrent Units (GRU). To overcome the **challenge** of uncertainty estimation in sequential deep models, our proposal directly quantifies uncertainty *deterministically*, without the need for costly sampling-based estimation. We show that while uncertainty is quite useful by itself in computer vision and machine learning, we also demonstrate that it can play a key role in enabling statistical analysis with deep networks in neuroimaging studies with normative modeling methods (Hwang et al., 2019b).

CH 6. Modeling the Relationship between Sequential Biomarker Modalities via Sequential Invertible Neural Networks

Understanding the progression pattern of various AD related pathology such as amyloid beta can be achieved by analyzing the longitudinal neuroimaging samples within data. For instance, by comparing the amyloid beta load over time measured by Pittsburgh Compound B (PiB) Positron Emission Tomography (PET) scans (Fig. 1.2a) between a risk-enriched group with a high risk of AD and a control group with a low risk of AD,

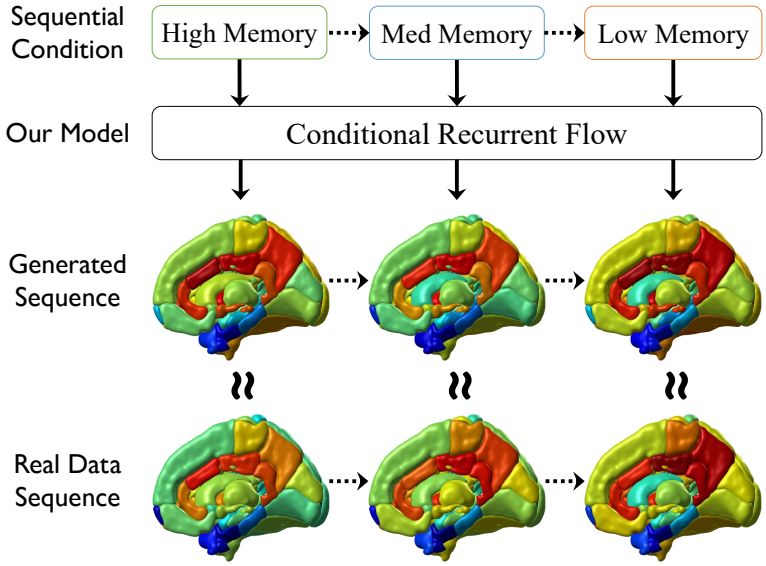


Figure 1.8: Main idea of Chapter 6: Conditional sequence generation. **1) Given:** a sequential condition of decreasing memory function (i.e., a memory test score sequence $\mathbf{y}_i^1 \rightarrow \mathbf{y}_i^2 \rightarrow \mathbf{y}_i^3$ indicating High→Medium→Low Memory performance). **2) Model:** Conditional Recurrent Flow (Our model). **3) Generate:** a sequence of brain image progression $\mathbf{x}_i^1 \rightarrow \mathbf{x}_i^2 \rightarrow \mathbf{x}_i^3$ corresponding to the given memory progression (i.e., brain regions with high (red) and low (blue) disease pathology). The Generated Sequence follows the trend of the Real Data Sequence (i.e., similar ($\approx$) to the real brain image progression) from the subjects with similarly decreasing memory scores.

we may be able to locate regions within the brain which show developmental differences. While various longitudinal neuroimaging datasets are growing, such collective effort from multiple AD research sites still require a substantial amount time (e.g., scans every 2 years $\Rightarrow$ 4 years for a 3 image sequence) and money (e.g., ~$4K per PiB PET scan). Such small sample size **challenge** in longitudinal neuroimaging (e.g., <300 subjects with 3 visits of PiB PET scans) often results in weak statistical analyses even with one of the largest public neuroimaging datasets. Further, for other related tasks such as longitudinal brain network prediction described in the previous chapter, successfully training similar types of sequential deep networks with millions of parameters are highly likely to be infeasible.

One promising solution is to generate additional realistic samples con-

ditioned on such "risk-enriched" and "control" conditions for more robust subsequent statistical analyses. For instance, existing generative models would train on existing data to learn the underlying distribution of the measurements (e.g., brain images) in latent spaces conditioned on covariates (e.g., cognition), and generate independent samples that are identically distributed in the latent space. Such models may work for cross-sectional studies, however, they are not suitable to generate data for *longitudinal* studies that focus on "progressive" behavior in a sequence of data such as a trajectory of pathologies we are interested in. In **Chapter 6**, we will present our conditional generative model for longitudinal data generation by designing a sequential invertible neural network which captures the mapping between two sequential modalities through *temporal latent representation* and generates sequences of brain image features conditioned on associated sequences of covariates by sampling from the mapped latent space. Fig. 1.8 illustrates the goal of this chapter.

CH 7. Retrospectively Understanding the Relationships between Alzheimer's Disease Pathologies in the Past

Understanding the early pattern of amyloid, a crucial biomarker of AD, has been a challenging task due to lack of subjects with early longitudinal amyloid PET scans (e.g., <60 years of age). There have been several studies by our collaborators and others (Koscik et al., 2019a; Bilgel et al., 2016) which suggest that the time of onset (TO) when amyloid accumulation crosses a critical threshold at an early age is believed to be one of the earliest signs of AD progression. However, TOs of a vast majority of the subjects are not actually observed in the data since acquiring the scans which are early enough to directly estimate the TOs is very few. Thus, a crucial component of this analysis requires a model which can accurately estimate the TO into the past give a series (or a single time point) of scans.

On this end, these prior studies make two compromises: (1) an arbitrary clustering approach to estimate the group-level TOs and (2) the averaging of amyloid measures across the brain regions. Considering how various regions of the brain can often accumulate varying degrees of pathology in different patterns among the subjects, a model which estimates the accumulation patterns for *each region* and *each subject* would allow us to ask more straightforward scientific hypotheses. In **Chapter 7**, we will describe our recent work on this scientific goal while addressing these issues via conditional generative model presented in Chapter 6.

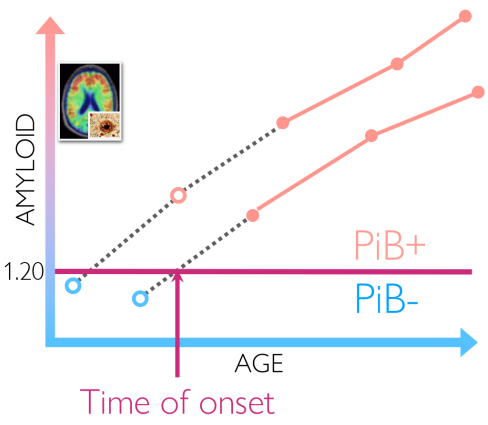


Figure 1.9: Main idea of Chapter 7: Given a series of PiB-DVR measures for a subject in multiple ROIs, the goal is to predict the past PiB-DVR trajectory retrospectively into the past. Then, the time of onset (TO) which the PiB-DVR accumulates over the threshold and becomes PiB+ is computed.

Finally, in **Chapter 8**, we will summarize the contributions of the thesis and discuss the future directions of our research.

In this preliminary chapter, we cover the basics of various latent representations that will be mentioned throughout the dissertation.

2.1 Eigenvalue Problem

Eigenvalue problems (Wilkinson, 1965) are ubiquitous in computer vision, covering a broad spectrum of applications. This will be an overview of the formulation of the eigenvalue problem and its generalized version along with the optimization problem to solve them which will appear in Chapter 4.

Specifically, for a square matrix $A \in \mathbb{R}^{n \times n}$, the eigenvalue problem is finding the eigenvectors $v_1, \ldots, v_n$ for $v_i \in \mathbb{R}^n$ and their corresponding eigenvalues $\lambda_1, \ldots, \lambda_n$ for $\lambda_i \in \mathbb{R}$ such that

$$Av_i = \lambda_i v_i \tag{2.1}$$

for all $i = 1, \ldots, n$. The matrix representation is

$$AV = V\Lambda \tag{2.2}$$

where $V = [v_1, \ldots, v_n] \in \mathbb{R}^{n \times n}$ is a matrix with eigenvector columns and $\Lambda = \mathtt{diag}([\lambda_1, \ldots, \lambda_n]) \in \mathbb{R}^{n \times n}$ is a diagonal matrix with eigenvalue diagonal entries.

The types of problems that we are interested in often involve a symmetric A (e.g., similarity matrix), in which case, its eigenvectors are orthonormal such that $V^T V = VV^T = I$ since $V^{-1} = V^T$. Further, when A is a symmetric positive semidefinite matrix (i.e., $x^T Ax \geqslant 0$ for any x), all its

eigenvalues λ_i are nonnegative. More generally, we see that

$$AV = V\Lambda \Rightarrow A = V\Lambda V^T \tag{2.3}$$

which is also called an eigenvalue decomposition or a spectral decomposition.

A specific use that we will see in Chapter 4 is spectral graph theory (Chung and Graham, 1997). Specifically, a graph G with n nodes where node i and j are connected by an undirected weighted edge w_{ij} can be encoded in an adjacency matrix $A \in \mathbb{R}^{n \times n}$ where $A(i,j) = w_{ij}$. When G is undirected, A is symmetric. The graph Laplacian L is then derived as follows:

$$L = D - A, \quad D(i,i) = \sum_{j=1}^{n} A(i,j) \tag{2.4}$$

where $D \in \mathbb{R}^{n \times n}$ is called the degree matrix where each of its diagonal entries $D(i,i)$ is the sum of total edge weights connected to node i. Now, L is a symmetric positive semidefinite matrix where the eigenvectors corresponding to lower order eigenvalues contain the "low frequency" information (i.e., the global structure of the graph) which reflects the latent structure of the graph Laplacian.

Generalized Eigenvalue Problem

The generalized eigenvalue problem (Parlett, 1998) involves another symmetric matrix $B \in \mathbb{R}^{n \times n}$ where the goal is to find eigenvectors $v_1, \ldots, v_n$ for $v_i \in \mathbb{R}^n$ and their corresponding eigenvalues $\lambda_1, \ldots, \lambda_n$ for $\lambda_i \in \mathbb{R}$ such that

$$Av_i = \lambda_i B v_i \tag{2.5}$$

for all $i = 1, \ldots, n$. The matrix representation is

$$AV = BV\Lambda \tag{2.6}$$

where $V = [v_1, \ldots, v_n] \in \mathbb{R}^{n \times n}$ is a matrix of eigenvectors and and $\Lambda = diag([\lambda_1, \ldots, \lambda_n]) \in \mathbb{R}^{n \times n}$ is a diagonal matrix of eigenvalues. The pair $\{A, B\}$ is also commonly referred to as the matrix pencil, and B is often called the mass matrix in some applications (e.g., structural mechanics (Bathe and Wilson, 1973)). In computer vision, the Normalized cut problem can also be formulated as a generalized eigenvalue problem (Shi and Malik, 2000). Note that when B is the identity matrix, this problem reduces to the standard eigenvalue problem. While B can be singular, it is often a positive definite matrix by construction in many applications.

Eigenvalue Optimization

Now we see how the eigenvalues and eigenvectors can be numerically computed. The following maximization optimization problem finds the eigenvector v that corresponds to the largest eigenvalue (conversely, minimum eigenvalue if it is a minimization problem) in Eq. (2.2):

$$\begin{aligned} \min_{v \in \mathbb{R}^n} \quad & v^T A v \\ \text{s.t.} \quad & v^T v = 1. \end{aligned} \tag{2.7}$$

We can easily see that a solutions is indeed an eigenpair since the Lagriangian for Eq. (2.7) is

$$\mathcal{L} = v^T A v - \lambda (v^T v - 1)$$

where λ is the Lagrange multiplier, and its derivative set equal to zero is

$$\frac{\partial \mathcal{L}}{\partial v} = 2Av - 2\lambda v = 0 \Rightarrow Av = \lambda v$$

which is equivalent to Eq. (2.1).

In practice, we are often interested in finding p eigenvectors $V \in \mathbb{R}^{n \times p}$

that correspond to the p largest eigenvalues which we can compute as

$$\begin{aligned} \min_{V \in \mathbb{R}^{n \times p}} & \operatorname{tr}(V^T A V) \\ \text{s.t.} \quad & V^T V = I \end{aligned} \tag{2.8}$$

where $\operatorname{tr}(\cdot)$ denotes the trace functional. Note that we can also find all the eigenpairs since the Lagriangian for Eq. (2.8) is

$$\mathcal{L} = \operatorname{tr}(V^T A V) - \operatorname{tr}(\Lambda^T (V^T V - I))$$

where Λ is a diagonal matrix with Lagrange multipliers, and its derivative set equal to zero is

$$\frac{\partial \mathcal{L}}{\partial V} = 2AV - 2V\Lambda = 0 \Rightarrow AV = V\Lambda$$

which is equivalent to Eq. (2.2).

Stiefel Manifold

The orthogonality constraint in Eq. 2.8 implies that the set of $n \times p$ orthonormal matrices that we optimize over is called the Stiefel manifold, and we discuss this concept briefly.

For vector spaces V and W, let $L(V, W)$ denote the vector space of linear maps from V to W. Thus, the space of $L(\mathbb{R}^N, \mathbb{R}^p)$ may be identified with the space $\mathbb{R}^{n \times p}$ of $n \times p$ matrices. An injective linear map $u : \mathbb{R}^n \to V$ is called a $n-$frame in V. Specifically, the set $GF_{n,p} = \{u \in L(\mathbb{R}^n, \mathbb{R}^p) : \operatorname{rank}(u) = n\}$ of $n-$frames in $\mathbb{R}^p$ is called the Stiefel manifold. As a special case, when $n = p$, $GF_{n,n}$:=GF_n is the General Linear group or the set of $n \times n$ matrices with nonzero determinant. In short, a Stiefel manifold is the set of $n \times p$ orthonormal matrices (with a Riemannian structure). The set of all $n-$dimensional (vector) subspaces $\alpha \subseteq \mathbb{R}^p$ is called the Grassmann

manifold of $n-$planes in $\mathbb{R}^p$ and denoted by $GR_{n,p}$. With these definitions, we see that the Grassmann manifold is just the Stiefel manifold quotiented by the Orthogonal group (set of orthogonal matrices) in $n-$dimensions.

Thus, Eq. 2.8 is actually an implicit optimization over the Grassmann manifold rather than the Stiefel manifold. This is because the objective function is invariant to a rotation in $\mathbb{R}^p$ of the decision variables, that is, replacing V with VQ so that $Q \in \mathbb{R}^{p\times p}$, $Q^TQ = I$, we have that,

$$\text{tr}((VQ)^TA(VQ)) = \text{tr}(Q^T(V^TAV)Q) = \text{tr}(V^TAV)$$

where the second equality is due to the similarity invariance property of the trace functional. For more details on these topics outside the scope of this dissertation such as an exponential map, tangent space, and retraction, (Absil et al., 2009) is a good source to start.

2.2 Tensor Decomposition

In this section, we provide some preliminaries of tensor decomposition which appears in Chapter 3. This is a higher-order generalization of the matrix factorization methods such as PCA that we have seen so far. For more details about tensor decomposition methods, the readers may refer to (Kolda and Bader, 2009).

Tensor

Tensors are essentially multidimensional or n-way arrays. Therefore a 0-way tensor is a scalar, a 1-way tensor is a vector (i.e., 1-dimensional array $\mathbf{X} \in \mathbb{R}^{I_1}$), a 2-way tensor is a matrix (i.e., 2-dimensional matrix $\mathbf{X} \in \mathbb{R}^{I_1\times I_2}$), and a 3-way tensor is a box-shaped 3-dimensional matrix ($\mathbf{X} \in \mathbb{R}^{I_1\times I_2\times I_3}$). This can be generalized to the n-way (or nth order or nth mode) tensor. For those with $n \geqslant 3$-ways, we call them higher-order tensors.

Thus, a n-way tensor has n indices, one for each dimension. A 3-way tensor, for instance, can be indexed with $i \in I_1$, $j \in I_2$, and $k \in I_3$ to select an entry $\mathbf{X}_{ijk} \in \mathbb{R}$. For every index that is fixed, the dimension of the corresponding subarray changes. For instance, if we fix the first two indices of $\mathbf{X}$, we get a vector $\mathbf{X}_{ij:} \in \mathbb{R}^{I_3}$ with the dimension corresponding to the last index. Similarly, fixing only one index results in a matrix (slice) $\mathbf{X}_{i::} \in \mathbb{R}^{I_2 \times I_3}$. This is equivalent to choosing rows or columns in a matrix by fixing one of the dimensions.

Several important definitions are also generalizable from those commonly used with matrices. Let an outer product of two vectors $\mathbf{u} \in \mathbb{R}^{I_1}$ and $\mathbf{v} \in \mathbb{R}^{I_2}$ be a matrix $\mathbf{X} = \mathbf{u}\mathbf{v}^T \in \mathbb{R}^{I_1 \times I_2}$. Simply, each element of $\mathbf{X}$ is just a product of the elements in each of the vectors such that $\mathbf{X}_{ij} = \mathbf{u}_i \mathbf{v}_j$. Thus, generalizing this to an outer product of n vectors $\mathbf{a}^{(1)}, \cdots, \mathbf{a}^{(n)}$ results in a n-way tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times \cdots \times I_n} = \mathbf{a}^{(1)} \otimes \cdots \otimes \mathbf{a}^{(n)}$ which is constructed as follows:

$$\mathbf{X}_{i_1 i_2 \cdots i_n} = \mathbf{a}^{(1)}_{i_1} \mathbf{a}^{(2)}_{i_2} \cdots \mathbf{a}^{(n)}_{i_n}. \tag{2.9}$$

In this case, $\mathbf{X}$ is a rank-1 tensor similar to how a matrix of an outer product of two vectors is also a rank-1 matrix. If a tensor is a sum of R rank-1 tensors scaled by λs such that

$$\mathbf{X} = \sum_{r=1}^{R} \lambda_i \mathbf{a}^{(1)}_r \otimes \cdots \otimes \mathbf{a}^{(n)}_r \tag{2.10}$$

where each $\mathbf{a}^{(i)}_r \in \mathbb{R}^{I_i}$ is a vector, then it is a rank-R tensor. The set of vectors for each order $\mathbf{A}_i = [\mathbf{a}^{(i)}_1, \cdots, \mathbf{a}^{(i)}_r] \in \mathbb{R}^{I_i \times r}$ is called a factor matrix, so a rank-r tensor can be decomposed into n factor matrices with the rank-1 scaling factors $\lambda_1, \ldots, \lambda_R$.

CP Decomposition

In practice, when a data can be constructed as a tensor, a common goal is to decompose it into a set of components (e.g., factor matrices) that are more explainable, similar to how principal components of PCA and singular vectors of SVD compactly characterize the overall data. Thus, analogous to the matrix decomposition methods, tensors are decomposed via various tensor decomposition methods.

We first describe CANDECOMP (canonical decomposition) (Carroll and Chang, 1970) and PARAFAC (parallel factors) (Harshman et al., 1970) which are essentially identical but independently developed, thus together called CANDECOMP/PARAFAC, or CP, decomposition. For now, we focus on 3-way tensors. The CP decomposition factorizes a tensor $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$ into a sum of rank-1 tensors best approximates the original tensor:

$$\mathbf{X} \approx \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r = [[\mathbf{A}, \mathbf{B}, \mathbf{C}]] \tag{2.11}$$

where R is the rank and $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$, and $\mathbf{c}_r \in \mathbb{R}^K$ are vectors for $r = 1, \ldots, R$, and $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_R]$, $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_R]$, $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_R]$ are the corresponding factor matrices. Again, this can be generalized to n-way tensors:

$$\mathbf{X} \approx \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \otimes \cdots \otimes \mathbf{a}_r^{(n)} = [[\mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(n)}]]. \tag{2.12}$$

Finding such decomposition with R that closely approximates the original tensor is called a low-rank approximation. To find such factor matrices (for a 3-way tensor $\mathbf{X}$), first we matricize $\mathbf{X}$ into three different forms:

$$\hat{\mathbf{X}}_{(1)} = (\mathbf{C} \odot \mathbf{B})\mathbf{A}^T, \quad \hat{\mathbf{X}}_{(2)} = (\mathbf{C} \odot \mathbf{A})\mathbf{B}^T, \quad \hat{\mathbf{X}}_{(3)} = (\mathbf{B} \odot \mathbf{A})\mathbf{C}^T \tag{2.13}$$

where $\odot$ is the Khatri-Rao product:

$$\mathbf{A} \odot \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1 \dots \mathbf{a}_K \otimes \mathbf{b}_K] \in \mathbb{R}^{(IJ)\times K}$$

for $\mathbf{A} \in \mathbb{R}^{I\times K}$ and $\mathbf{B} \in \mathbb{R}^{J\times K}$. Then, we can perform an Alternating Least Squares (ALS) algorithm to iteratively solve for each of Eq. (2.13) such that

$$\begin{aligned}
\mathbf{A} &\leftarrow \arg\min_{\mathbf{A}} \|\mathbf{X}_{(1)} - (\mathbf{C} \odot \mathbf{B})\mathbf{A}^T\|_2^2 \\
\mathbf{B} &\leftarrow \arg\min_{\mathbf{B}} \|\mathbf{X}_{(2)} - (\mathbf{C} \odot \mathbf{A})\mathbf{B}^T\|_2^2 \\
\mathbf{C} &\leftarrow \arg\min_{\mathbf{C}} \|\mathbf{X}_{(3)} - (\mathbf{B} \odot \mathbf{A})\mathbf{C}^T\|_2^2
\end{aligned}$$

over several iterations until convergence.

Tucker Decomposition

The Tucker decomposition (Tucker, 1966) consists of a core tensor $\mathbf{G} \in \mathbb{R}^{P\times Q\times R}$ that the factor matrices perform a specific type of product called the m-mode product. Specifically, the m-mode product of a tensor $\mathbf{X} \in \mathbb{R}^{I_1\times\cdots\times I_n}$ and a matrix $\mathbf{A} \in \mathbb{R}^{J\times I_m}$ is

$$\mathbf{Y} = \mathbf{X} \times_m \mathbf{A} \in \mathbb{R}^{I_1\times\cdots I_{m-1}\times J\times I_{m+1}\times\cdots\times I_n} \tag{2.14}$$

where each element is

$$\mathbf{Y}_{i_1\cdots i_{m-1} j i_{m+1}\cdots i_n} = \sum_{i_m=1}^{I_m} \mathbf{X}_{i_1\cdots i_n}\mathbf{a}_{j i_m} \tag{2.15}$$

which can essentially be thought of as performing a tensor-matrix multiplication along the mth mode, effectively transforming the mode's dimension from I_m to J. Thus, if $\mathbf{A}$ in Eq. (2.14) is a vector with $J = 1$, $\mathbf{Y} = \mathbf{X} \times_m \mathbf{A}$ effectively loses its mth mode.

Now, the Tucker decomposition constructs the factor matrices where they can have different ranks such that $\mathbf{A} \in \mathbb{R}^{I \times P}$, $\mathbf{B} \in \mathbb{R}^{J \times Q}$, and $\mathbf{C} \in \mathbb{R}^{K \times R}$ as follows:

$$\mathbf{X} \approx \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr} \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r = \mathbf{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}. \tag{2.16}$$

This can be solved using the ALS algorithm as well based on the following matricization:

$$\hat{\mathbf{X}}_{(1)} = (\mathbf{C} \odot \mathbf{B})(\mathbf{A}\mathbf{G}_{(1)})^T, \quad \hat{\mathbf{X}}_{(2)} = (\mathbf{C} \odot \mathbf{A})(\mathbf{B}\mathbf{G}_{(1)})^T, \quad \hat{\mathbf{X}}_{(3)} = (\mathbf{B} \odot \mathbf{A})(\mathbf{C}\mathbf{G}_{(1)})^T \tag{2.17}$$

where $\mathbf{A}_{(m)}\mathbf{B}$ is a short for $\mathbf{A} \times_m \mathbf{B}$. Again, the Tucker decomposition is also generalizable to n-way tensors.

There exist other tensor decomposition methods depending on the structure of the tensor. A specific type of data that we will see in Chapter 3 is the multi-relational data where each slice of a tensor describes a relationship between problem-specific objects. The latent representations derived from these methods have solved various challenging tasks including community detection (Papalexakis et al., 2013), visual relationship learning (Hwang et al., 2018), and word representation learning (Jenatton et al., 2012).

2.3 Sequential Deep Neural Network

Preliminaries of sequential deep neural network in this section will be beneficial to digest its variants which appear in Chapter 5 and Chapter 6. For more details about the empirical evaluations of RNN and its variants, the readers can refer to (Jozefowicz et al., 2015).

Neural Network

In machine learning, artificial neural network (neural network) is a type of computational model which its structure is inspired by the biological neural networks in the brain (McCulloch and Pitts, 1943). The most basic neural network model is a perception (Rosenblatt, 1958) (Fig. 2.1a). Specifically, given an input $\mathbf{x} \in \mathbb{R}^n$, the model consists of weights $\mathbf{w} \in \mathbb{R}^n$ such that the output $o \in \{0, 1\}$ is computed as follows:

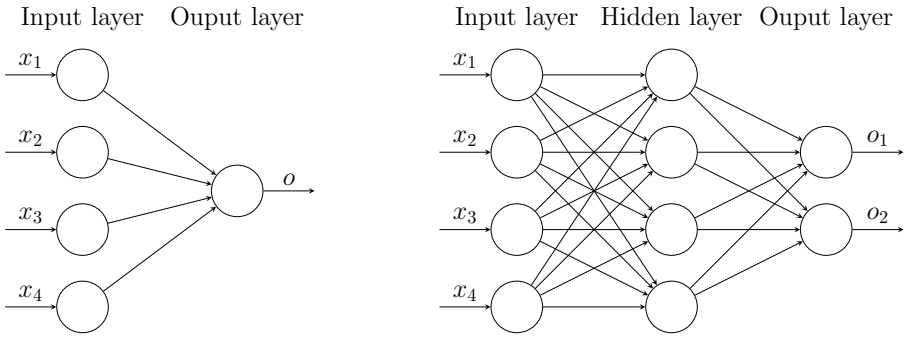
$$o = \begin{cases} 1 & \text{if } \mathbf{w}_0 + \sum_{i=1}^{n} \mathbf{w}_i \mathbf{x}_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.18}$$

where $\mathbf{w}_i$ is the ith element of $\mathbf{w}$. The goal is to find the weights that correctly predict the output for a set of samples/instances in data. This is achieved by an iterative training process: given a set of training instances of input $\mathbf{x} \in \mathbb{R}^n$ and output $\mathbf{y} \in \mathbb{R}$, we (1) randomly initialize weights, and (2) iterate through the training instances until convergence by updating each weight $\mathbf{w}_i \leftarrow \mathbf{w}_i + \delta\mathbf{w}_i$ where $\delta\mathbf{w}_i = \eta(y - o)\mathbf{x}_i$ and η is a learning rate. A nonlinear activation such as a sigmoid function is commonly performed on each output o to impose a nonlinear transformation. Essentially, a single layer fully-connected network is a function which given an input $\mathbf{x} \in \mathbb{R}^{n_1}$, its network parameters (weights) are a transformation matrix $\mathbf{W} \in \mathbb{R}^{n_2 \times n_1}$ followed by an activation function f as follows:

$$\mathbf{y} = f(\mathbf{W}\mathbf{x}) \tag{2.19}$$

where we assume the bias term $\mathbf{w}_0$ is a part of $\mathbf{W}$ without loss of generality.

Multi-layer neural networks have been studied where the outputs of the first layer become the subsequent hidden layer which acts as the inputs to the next set of operations and so on (Fig. 2.1b). A multi-layer neural



(a) Single-layer Neural Network (b) Multi-layer Neural Network

Figure 2.1: Examples of Neural Networks

network with L layers can essentially be formulated as follows:

$$\mathbf{y} = f(\mathbf{W}_L \dots f(\mathbf{W}_2(f(\mathbf{W}_1\mathbf{x}))) \dots) \tag{2.20}$$

where the weights $\mathbf{W}_1, \dots, \mathbf{W}_L$ could have varying dimensions.

To efficiently learn the weights, the backpropagation algorithm (Werbos, 1974) is deployed in almost all cases for training deep neural networks. Let us briefly described the procedure for a simple case. To update a weight vector $\mathbf{w}$ for a single layer network, the backpropagation algorithm first computes the error that $\mathbf{w}$ contributes in terms of a sum of squared error loss function between the ground truth $\mathbf{y} \in \mathbb{R}^n$ and the prediction $\mathbf{o} \in \mathbb{R}^n$ given a sample:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}(y_i - o_i)^2. \tag{2.21}$$

Then, the direction and the magnitude of $\mathbf{w}$ contributing to the error is computed by finding the gradient $\nabla E(\mathbf{w})$ in the weight space:

$$\nabla E(\mathbf{w}) = [\frac{\partial E}{\partial \mathbf{w}_0}, \frac{\partial E}{\partial \mathbf{w}_1}, ..., \frac{\partial E}{\partial \mathbf{w}_n}]. \tag{2.22}$$

Next, the current $\mathbf{w}$ is updated in the direction that minimizes the error

incurred with respect to $\mathbf{w}$ scaled by some small learning rate $\eta > 0$

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla E(\mathbf{w}). \tag{2.23}$$

In practice, we repeat this procedure until the training does not make a meaningful progress from small gradients (e.g., $|\nabla E(\mathbf{w})| < \epsilon$ for some small $\epsilon > 0$).

Recurrent Neural Network (RNN)

Given a sequential sample $\mathbf{x}_1, \ldots, \mathbf{x}_T$ where each $\mathbf{x}_t \in \mathbb{R}^n$ is a vector, a typical neural network may not be feasible for several reasons. From the technical perspective, the overall input feature dimension could be varying (i.e., T could vary) so the models with a fixed structure (e.g., multi-layer neural network) are hard to utilize directly. From the application perspective, it may be beneficial to explicitly capture the sequential pattern that could be generalized beyond the given observation into the future time points (i.e., for $t > T$). A recurrent neural network (RNN) addresses both of these issues. Formally, a RNN recursively incorporates the output from the previous time point as a part of the input along with $\mathbf{x}_t$:

$$\mathbf{o}_t = f(\mathbf{W}[\mathbf{o}_{t-1}, \mathbf{x}_t]) \quad \text{for } t = 1, \ldots, T \tag{2.24}$$

where $[\mathbf{o}_{t-1}, \mathbf{x}_t]$ is a vector concatenation. This could be viewed as a T-layered neural network where the weights across the T layers are shared (i.e., $\mathbf{W}$ is shared for all t). In practice, for some sequential problem formulations, each intermediate output $\mathbf{o}_t$ could be an input to another network (i.e., $\mathbf{y}_t = g(\mathbf{o}_t)$) at each time point to produce a sequential output. This recurrent setup also allows us to recursively predict subsequent time points for $t > T$ by using the output prediction $\mathbf{y}_t$ as an (estimated) input to the next time point (i.e., $\hat{\mathbf{x}}_{t+1} \leftarrow \mathbf{y}_t$). This recursive procedure, however,

with the shared $\mathbf{W}$ causes the vanishing/exploding gradient problem where the gradients computed with respect to the same $\mathbf{W}$ repeatedly quickly vanish/explode over a series computations (Chung et al., 2014). In other words, the RNN's ability to learn the sequential patterns from the past becomes less optimal for long sequences since the errors from a distant history cannot be learned effectively. While there exist several solutions such as the truncated backpropagation algorithm and sequence partitioning, variants of RNNs have been developed to explicitly address these issues. We will describe two main variants next.

Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) is a variant of RNN with a special structure that explicitly performs several mechanisms for learning long sequences while addressing the vanishing/exploding gradient issues. Specifically, each cell of LSTM consists of multiple gates with distinct functions. Each gate is essentially a neural network itself with a weight matrix: $\mathbf{W}$ for $[\mathbf{h}_{t-1}, \mathbf{x}_t]$ which is a concatenation of a hidden state vector $\mathbf{h}_{t-1}$ from the previous time point and the current input vector $\mathbf{x}_t$. An LSTM defines the gates and states as follows:

$$
\begin{aligned}
\text{Forget Gate: } \mathbf{f}_t &= \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t]) \\
\text{Input Gate: } \mathbf{i}_t &= \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t]) \\
\text{Output Gate: } \mathbf{o}_t &= \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t]) \\
\text{Cell State: } \mathbf{c}_t &= \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tanh(\mathbf{W}_c[\mathbf{h}_{t-1}, \mathbf{x}_t]) \\
\text{Hidden State: } \mathbf{h}_t &= \mathbf{o}_t \circ \tanh(\mathbf{c}_t)
\end{aligned}
$$

where $\circ$ is an element-wise product and tanh is a hyperbolic tangent activation function. Intuitively, $\mathbf{f}_t$ is a vector ranging from 0 to 1 which acts as a gate to "forget" certain entries of the previous cell state vector $\mathbf{c}_{t-1}$

(i.e., $\mathbf{f}_t \circ \mathbf{c}_{t-1}$). Similarly, $\mathbf{i}_t$ is a vector which keeps the "desired" entries from the current cell state vector candidate which is $\tanh(\mathbf{W}_c[\mathbf{h}_{t-1}, \mathbf{x}_t])$. The new cell state $\mathbf{c}_t$ then is used to adjust the output vector $\mathbf{o}_t$ to produce a hidden state vector $\mathbf{h}_t$ for the next time point. These components are interconnected in each LSTM cell, and it has been one of the most popular RNN variants for its ability to retain long term information through long sequences.

Gated Recurrent Units (GRU)

Another popular RNN variant is Gated Recurrent Units (GRU) (Chung et al., 2014). The overall cell structure resembles that of LSTM, but a GRU does not represent the cell state and hidden state separately. Specifically, its updates take the following form:

$$\begin{aligned}
\text{Reset Gate: } \mathbf{r}_t &= \sigma(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \mathbf{h}_{t-1}) \\
\text{Update Gate: } \mathbf{z}_t &= \sigma(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \mathbf{h}_{t-1}) \\
\text{State Candidate: } \hat{\mathbf{h}}_t &= \tanh(\mathbf{W}_{\hat{h}} \mathbf{x}_t + \mathbf{U}_{\hat{h}}(\mathbf{r}_t \circ \mathbf{h}_{t-1})) \\
\text{Cell State: } \mathbf{h}_t &= (1 - \mathbf{z}_t) \circ \hat{\mathbf{h}}_t + \mathbf{z}_t \circ \mathbf{h}_{t-1}
\end{aligned}$$

where $\mathbf{W}_{\{r,z,\hat{h}\}}$ and $\mathbf{U}_{\{r,z,\hat{h}\}}$ are the weights for their corresponding updates and vectors. The main simplification comes from how $\mathbf{h}_t$ functions as both a cell state vector and a hidden state vector. Due to its relatively simpler formulation without sacrificing the performance, GRU has been another popular RNN variant for various applications (Chung et al., 2014; Zhao et al., 2017).

3 ROBUST VISUAL RELATIONSHIP LEARNING

The first type of relationship we focus on in the thesis is found in natural *images*. This chapter tackles a computer vision problem where the goal is to understand images as humans do (e.g., constructing higher-level descriptions about the objects such as "*the glass is on the table*" in Fig. 1.1) which is not a straightforward task. To achieve this, we will construct the latent representations of the *visual relationships of objects* in images.

3.1 Overview

The core primitives of an image, are the objects and entities that are captured in it. As a result, a strong thrust of research, formalized within detection and segmentation problems, deals with accurate identification of such entities, given an image. On the other hand, there is consensus that to "understand" an image from a human's perspective (Lu et al., 2016; Johnson et al., 2015), higher-level cues such as the relationship between the objects are critical. Being able to reason about which entities are likely to *co-occur* (Mensink et al., 2014; Ladicky et al., 2010) and *how they interact* (Yao and Fei-Fei, 2010; Deng et al., 2014) is a powerful mid-level feature that endows a system with auxiliary information far beyond what individual object detectors provide. Starting with early work on AND-OR graphs (Lu et al., 2014b; Li et al., 2016) and logic networks (Tran and Davis, 2008; Song et al., 2013), algorithms which make use of relational learning are becoming mainstream within vision, offering strong performance on categorization and retrieval tasks (Alberti et al., 2014; Desai and Ramanan, 2012). Furthermore, many interesting applications (Chandrasekaran et al., 2016; Wu et al., 2016a; Antol et al., 2015) have begun to appear as richer datasets become available (Antol et al., 2015; Lu et al., 2016; Zhu et al., 2016; Xu et al., 2017).

Figure 3.1: Examples of visual relationships detected by our algorithm given objects and their object bounding boxes. The left two relationships (**green box**) were observed in the training set. The right three relationships (**orange box**) *not* observed in the training set are potentially much harder to detect.

Let us consider the process of setting up a corpus of data to precisely characterize the intuition above. Given a sufficiently large set of images where the objects have been localized (e.g., via human supervision), we process the images and specify the "relationship" between the objects; for example, person and couch related by sitting on and/or person and bike related by riding. Then, with a learning module in hand, say which extracts the latent representations, it should be possible to *learn* these associations to facilitate concurrent estimation of the object class *as well as* their relationship. For instance, a model may suggest that given a high confidence for the bike class, a smaller set of classes for the other object are likely, and perhaps, a small set of relationships may explain the semantic association between those two objects. Naturally, even when the 'base' set of relationships is small, such a construction can help object/relationship detection. The authors in (Sadeghi and Farhadi, 2011) showed that this idea of "Visual Phrases" performs well even when provided with a small set of 13 common relationships. However, as one may expect, for such a learning task to work, the training data size should be sufficient to cover all possible relationships. But as we make the universe of relationships richer, the distribution of relationships becomes skewed due to their infrequency. Also, a challenge is the availability of a large dataset that will enable the learned model to be transferable to other images in the wild.

In 2016, (Krishna et al., 2016) presented a visual relationship dataset,

Visual Genome, to help research on this topic: over 100K images with 42K unique relationships. Visual Genome is a massive expansion of the Scene Graph dataset (Johnson et al., 2015) (gives an image as a first-order network of its objects (vertices) and their visual relationships (edges)). Visual Genome connects the individual scene graphs to one another based on their common objects and/or relationships encoding the inter-connectedness of many complex object interactions. These two datasets are the starting points for our proposed algorithm.

From Visual Phrases to Scene Graph Prediction. Given a set of detected objects (i.e., person, dog, phone objects) in an image and possible predicates (i.e., on, next to, hold predicates), the goal is to infer the most likely relationships (i.e., ⟦person, hold, phone⟧ relationship) among the objects, see Fig. 3.1. The Visual Phrases based algorithm (Sadeghi and Farhadi, 2011) builds a model for *each* unique relationship instance to fully detect all possible relationships, i.e., # of predicates $\times$ # of object categories^2. Independent object-wise predictions are combined using a decoding scheme that takes all responses and then decides on the final image-specific outcome. The formulation is effective but as noted by (Lu et al., 2016), it becomes infeasible as the number of unique relationships (⟦object, predicate, object⟧) exceed several thousands – as is the case in the new Visual Genome dataset. To address this limitation, in (Lu et al., 2016) the authors propose building "joint" models that do *not* enumerate the set of all relationships and instead are proportional to the number of object categories plus predicates. This set is much smaller and effective to the extent that these fewer degrees of freedom capture the large number of relationships. As discussed in (Lu et al., 2016), often the language prior can compensate for such disparity between the model complexity and dataset complexity but also suffers if the semantic word embeddings fall short (Atzmon et al., 2016). Recently, as a natural extension to the individual relationship detection, understanding an image at a broader scope as a *scene graph* (Xu et al., 2017) has been pro-

posed where the goal is to infer the entire interconnectedness of the objects (nodes) in the image with various visual relationships (edges). While the detection on objects and relationships 'help' each other, relatively more challenging visual relationship inference is often the bottleneck within such combined approaches.

Key Components for Improved Visual Relationship Learning. A hypothetical model may offer improvements in visual relationship learning if it has the following properties: **(1)** Leaving aside empirical issues, the model complexity (i.e., degrees of freedom) should be able to compensate for the complexity of the data (i.e., number of object categories) while still guaranteeing performance gains for the core learning problem under mild assumptions. **(2)** Additionally, it would be desirable if the above characteristic can also generalize to unseen data (i.e., relationships *not* in training data) with little information about unseen observations (i.e., *unknown* category distributions). Our model offers these nice qualities to derive a regularization for use within any visual relationship detection pipeline.

Contributions

In this chapter, we provide the following contributions: **(i)** We view visual relationship learning as a slightly adapted instantiation of a multi-relational learning model. Despite its non-convex form, we show how recent results in linear algebra yield an efficient optimization scheme, with some guarantees towards a solution. **(ii)** We derive sample complexity bounds which demonstrate that despite the ill-posed nature, under sensible conditions, inference can indeed be performed. This scheme yields powerful visual relationship priors despite the extremely sparse nature of the data. Empirically, we consistently improve visual relationship prediction over the best known results (Lu et al., 2016) on Scene Graph. **(iii)** Our proposal integrates the priors with an adaptation of visual relationship detection architecture. This end-to-end construction brings the best per-

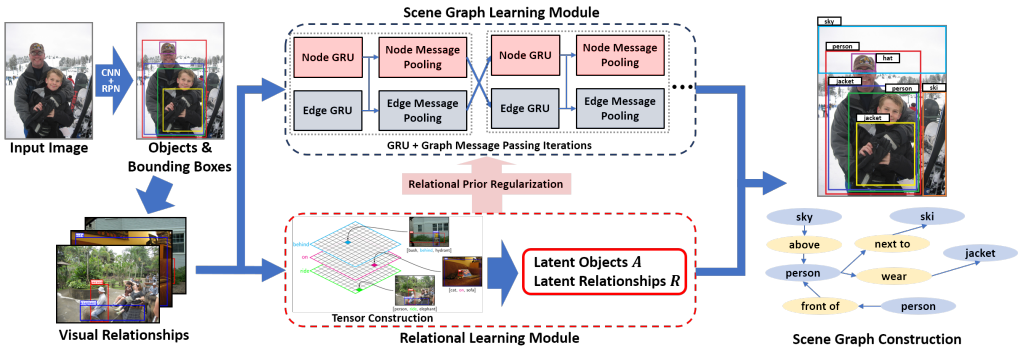


Figure 3.2: An end-to-end scene graph detection pipeline. In training, (left) given an image, its initial object bounding boxes and relationships are detected. Then, (top middle) its objects and relationships are estimated via scene graph learning module (Xu et al., 2017). Our tensor-based relational module (top bottom) provides a visual relationship prediction as a dense relational prior to refine the relationship estimation which also regulates the learning process of the scene graph module. In testing, (right) the scene graph of an image is constructed based on both modules.

formance of the much more challenging scene graph prediction tasks (Xu et al., 2017) on the Visual Genome dataset by modulating the deep neural network structure with a provably stable relational learning module. The key leverage comes from overcoming the sparsely observed visual relationships (~2% of possible relationships) with contribution (i)-(ii).

3.2 Relational Learning in Vision

In this section, we briefly review some of the related works. In the past years, low-to-mid level computer vision tasks have seen a renaissance leading to effective algorithms (Kulchandani and Dangarwala, 2015; Patel et al., 2015) and various datasets (Lin et al., 2014; Antol et al., 2015; Zhu et al., 2016). Building upon these successes, higher-level tasks, such as scene understanding (Eslami et al., 2016; Zitnick et al., 2016) and relationship inference (Lebeda et al., 2015; Lu et al., 2016; Wang et al., 2016b; Xu et al., 2017), which often rely on the lower-level modules are being more intensively studied. In particular, inferring the *visual relationship* between

objects is the next logical goal – going from object level detection to semantic relations among objects for higher-level relationships. For instance, simple contextual features such as co-occurrence (Ladicky et al., 2010; Mensink et al., 2014) are useful but not rich enough for detailed semantic relationship among objects such as those required within VQA (Antol et al., 2015). On the other hand, human-specific relationships based on human-object interaction (Rohrbach et al., 2013; Yao and Fei-Fei, 2010), while expressive, limit the scope of information inferable from natural images containing many types of objects. From a different perspective, inferring visual information from images under various assumptions (i.e., in the wild) has been utilized to retrieve task-specific visual information as well (Ramanathan et al., 2015; Thomason et al., 2014).

A deeper understanding of images is being successfully demonstrated in various semantic inference tasks. For instance, answering abstract questions related to a given image called visual question answering (VQA) (Antol et al., 2015) has shown good results (Zhu et al., 2016; Andreas et al., 2016) with the availability of various datasets (Antol et al., 2015; Zhu et al., 2016). Also, image captioning (Chen and Lawrence Zitnick, 2015; Xu et al., 2015) can infer detailed high-level knowledge from image.

In this chapter, we focus on inferring a mid/high level description commonly referred to as *visual phrases* (Sadeghi and Farhadi, 2011; Krishna et al., 2016) that provides systematically structured visual relationships (i.e., person rides a car as $[\![\text{person, ride, car}]\!]$) that is both quantifiable and expressive (i.e., person related to car by predicates ride and next to). Conversely, if precise visual phrases are provided, valuable high-level relationship information can also be passed down in a top-down manner for useful lower-level task like object recognition (Choi et al., 2013; Sadeghi and Farhadi, 2011).

For instance, understanding an image in terms of the objects *and* their visual relationships has been recently formulated as a *scene graph detection*

(Xu et al., 2017) based on the large-scale Visual Genome dataset (Krishna et al., 2016) which requires simultaneously performing both higher-level visual phrase inference and lower-level object recognition. As seen on the right of Fig. 3.2, a successfully constructed scene graph provides rich context about the image for an upstream system-level model (i.e., captioning or VQA). Naturally, such type of inference demands solid performance from both relationship and objects detections, but the bottleneck often comes down to the difficulty of understanding visual relationship involving semantic ambiguities and sparse sample observations.

3.3 Collective Learning on Multi-relational Data

Much of our technical development will focus on distilling the sparsely observed relationship data towards a precise regularization that will be integrated into an end-to-end pipeline. Note that in Chapter 6, we will tackle this problem with a data generation technique to deal with sparsely observed relationship samples between sequential modalities. To setup our presentation for deriving this prior, we first briefly describe encoding/representing the data and then obtain an objective function to model the inference task for the Relational learning module in Fig. 3.2.

Tensor Construction. Suppose we are given a dataset of N images that contains n object categories and m possible predicates which are both indexed. For instance, an image can have an object $i \in \{1, \ldots, n\}$ having a predicate $k \in \{1, \ldots, m\}$ with another object $j \in \{1, \ldots, n\}$. We can construct a relationship tensor $X \in \mathbb{R}^{n \times n \times m}$ where $X(i, j, k)$ contains the number of occurrences of the i'th object and j'th object having k'th predicate in the dataset. If the relationship of person (object index i) and bike (object index j) described by ride (predicate index k) has shown up p times, then we assign $X(i, j, k) = p$. We can also think of X as a stack of m matrices

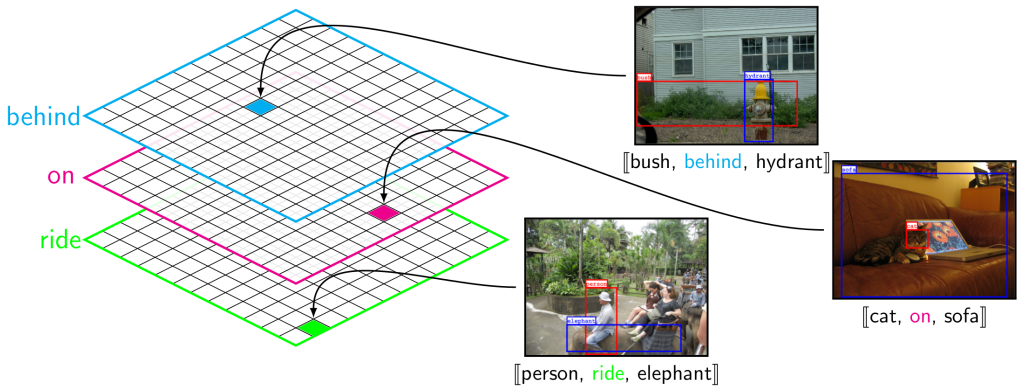


Figure 3.3: Muti-relational tensor $X \in \mathbb{R}^{n \times n \times m}$ given n object categories and m possible predicates. The value at $X(i, j, k)$ is the number of ⟦i'th object, k'th predicate, j'th object⟧ instances observed in the training set. Due to the sparse nature of the relationship instances, only ~ 1% of the tensor constructed from our training set has non-zero entries.

$X_k \in \mathbb{R}^{n \times n}$ for $k \in \{1, \ldots, m\}$: each X_k contains information about the k'th predicate among all the objects in the data (see Fig. (3.3)). Note that in practice, only a small fraction (i.e., ~1%) of the possible relationships are observed out of mn^2 possible relationships; the relationship tensor is extremely sparse.

Why Tensor Construction? In multi-relational learning such as visual relationship learning, it is critical to appropriately represent the interconnectedness of the objects (Singh and Gordon, 2008; Getoor et al., 2001). Such multi-relational information of any order can be easily encoded as a higher order tensor where its construction does not require any priors (parametric distributions in Bayesian Networks (Friedman et al., 1999)) or assumptions (Markov Logic network structure (Richardson and Domingos, 2006)). Our main motivation is: even though the objects are represented as points in $\mathbb{R}^n$, due to the sparse matrix slices X_k's, we may assume that the objects are embedded in fewer dimensions $r < n$. In principle, this can be accomplished by a message passing module (Xu et al., 2017) within the pipeline shown in Fig. 3.2 but experimentally, we find that concurrently learning both modules is challenging.

Why not Tensor Decomposition? Recently, many authors (Anandkumar et al., 2014; Hsu and Kakade, 2013) have shown that learning latent representations correspond to decomposing a tensor into low-rank components. While many standard techniques (Harshman and Lundy, 1994; Tucker, 1966) exist, they are inappropriate for multi-relational learning for a few reasons. For instance, polyadic decomposition (Harshman and Lundy, 1994) puts rigid constraints on the relational factor (i.e., diagonal core tensor) which is counterintuitive in relational learning (Nickel et al., 2011). Ideally, we want the *converse* construction where the relational factors are flexible with respect to the "latent" object representations. In that sense, our model is similar to the less widely used Tucker 2−decomposition (Tucker, 1966), but Tucker 2 allows too many degrees of freedom on the objective factor. Second, the solution of typical solvers (Harshman and Lundy, 1994; Tucker, 1966) is often not unique. This is not relevant in many factor analysis tasks that do *not* rely on the representations (i.e., Eigenfaces (Turk and Pentland, 1991)), but this property is undesirable in our formulation where we explicitly consider the relationships among the objects in their "latent" representations. In other words, two equally optimal solutions could interpret the same relationship differently. Thus, we need to impose consistency in representations by identifying a unique solution (via additional regularization).

In this section, we describe a novel relational learning algorithm which addresses the above issues and provides the generalization power needed for visual relationship detection. We first explain our model motivated by a three-way collective learning model (Nickel et al., 2011) which derives a set of latent object representations connected by relational factors. Later, we extend this formulation and describe our relationship inference model which guarantees a unique solution for consistent objects representations and their relationships. We then empirically show how our pipeline (Fig. 3.2) integrating the regularization (or prior) obtains benefits.

Three-way Relational Learning

Recall our mild assumption that the objects can be represented in a lower dimensional space with dimensions $r < n$. We will now explain our model in two steps: first, given the multi-relationship tensor X, our goal is to derive the latent representation of its objects $A \in \mathbb{R}^{n \times r}$ of rank r; secondly, assuming that we know the lower dimensional representation A of the objects, now we can define the relationship-specific factor matrix $R_k \in \mathbb{R}^{r \times r}$ for each $k \in \{1, \ldots, m\}$ for each relationship matrix X_k. Observe that A is common across all the relationships where the i'th row of A is the latent representation of the i'th object as desired. On the other hand, each factor matrix R_k individually corresponds to the k'th relationship and constitutes its respective matrix X_k (see Fig. (3.3)) with the common latent representation A. We can now write our model as,

$$X_k \approx A R_k A^T. \tag{3.1}$$

Hence, our optimization problem to solve is,

$$\min_{A, R_k} \sum_{k=1}^{m} \|X_k - A R_k A^T\|_F^2 \tag{3.2}$$

where we will learn A and R_k's simultaneously. Such a decomposition of a three dimensional tensor is referred to as Tucker 2−decomposition (Kolda and Bader, 2009). The "2" refers to the fact that we are learning two "types" of matrices in some sense.

Now we discuss a crucial property of the tensor X that is very relevant. Observe that since a relationship and its converse (i.e., person on bike and bike on person) need *not* always occur together, each X_k is not always symmetric, thus preventing us from effectively using many readymade tools from matrix analysis like the spectral theorem, eigendecomposition and so on. In our multi-relational tensor X, a predicate often cannot be sensibly applied in the other direction. Thus, we propose alternative strategies that includes certain reformulations. Before we present our final

algorithm to solve problem (3.2), we will show how certain reformulations will enable us to design efficient algorithms.

A possible solution strategy to solve the above formulation (3.2) is using a conventional approach such as the Alternating Least Squares (ALS) method (Carroll and Chang, 1970). In this method, one variable is optimized while fixing all the other variables. *Importantly, for the ALS algorithm to be efficient, we need all of the optimization subproblems to be easily solvable.* However, note that solving for A while fixing R_k's is not easy since it involves fourth order polynomial optimization.

3.4 Algorithm

In this section, we present our algorithm (Alg. 1) consisting of a novel initialization scheme followed by an iterative scheme to solve our multi-relational problem 3.2 with an additional regularization term that is weakly derived from (Tu et al., 2015). Then, we show how the algorithm can be integrated into the formulation in Fig. 3.2 as the Relational learning (RL) module which provides a dense predicate prior.

Multi-relational Tensor Factorization

To make our analysis easier, as the first step, we use auxiliary variables to decouple A and A^T in the objective function resulting in a method of multipliers type formulation,

$$\min_{A,R_k} \sum_{k=1}^{m} ||X_k - B_k A^T||_F^2 \quad \text{s.t.} \quad B_k = AR_k. \tag{3.3}$$

For the purpose of designing an algorithm, let us analyze only the objective function in Eq. (3.3) ignoring the equality constraints resembling matrix

factorization by letting $m = 1$:

$$\min_{A,B} ||X - BA^T||_F^2. \tag{3.4}$$

It is easy to see that the above problem can be solved exactly using the Singular Value Decomposition (SVD) of X. When $m > 1$, we need to identify matrix factorization type models where SVD (or something related) serves as a subroutine. Recent works use SVD as a subroutine in primarily a few different ways to solve problems that can be posed as matrix factorization problems: preprocessing step (Boutsidis and Gallopoulos, 2008) at each iteration (Jain et al., 2010) and thresholding schemes (Bansal et al., 2014). Intuitively, in the above works, the SVD of an *appropriate* matrix (chosen specifically depending on the problem context) provides a good estimate of the global optimal solution of rank constrained optimization problems both theoretically (Sanghavi et al., 2017) and practically in vision applications (Lu et al., 2014a). Essentially, these works show that with a specially constructed matrix, having an initialization already gets close to optimal solutions, and then any descent method is guaranteed to work. Unfortunately, these results do *not* extend to our case when $m > 1$. we generalize this idea, derive sample complexity bounds on the number of predicates needed to learn the latent representations and give an efficient algorithm.

Low-rank Initialization via SVD

For a given generic $X \in \mathbb{R}^{n \times n}$, Eq. (3.4) can be solved by

$$A = V\Sigma^{1/2}, \quad B = U\Sigma^{1/2} \tag{3.5}$$

where $U\Sigma V^T = X$ is the SVD of X. Under certain conditions, recent works such as (Sun and Luo, 2015) and (Tu et al., 2015) have shown that an

Algorithm 1 Alternating Block Coordinate Descent on (3.12)

Given: $X \in \mathbb{R}^{n \times n \times m}$, $X_k := X(:,:,k)$, rank $r > 0$
Low-rank Initialization:
$\bar{X} \leftarrow \sum_{k=1}^{m} X_k$
$\bar{U}\bar{\Sigma}\bar{V}^T \leftarrow \text{SVD}(\bar{X}, r)$
$A \leftarrow \bar{V}\bar{\Sigma}^{1/2}$
for $k = 1, \ldots, m$ **do**
    $B_k \leftarrow \bar{U}\bar{\Sigma}^{1/2}$
    $R_k \leftarrow (A^TA)^{-1}(A^TX_kA)(A^TA)^{-1}$
end for
Iterative descent method:
while Convergence criteria not met **do**
    $A \leftarrow$ gradient descent on (3.12) w.r.t. A
    for $k = 1, \ldots, m$ **do**
        $B_k \leftarrow$ gradient descent on (3.12) w.r.t. B
        $R_k \leftarrow \left(A^TA\right)^{-1}\left(A^TB_k\right)$
    end for
end while
Output: $A \in \mathbb{R}^{n \times r}$, $B_k \in \mathbb{R}^{n \times r}$, $R_k \in \mathbb{R}^{r \times r}$ for $\forall k$

initial point for other common low-rank decomposition formulations can be estimated within the "basin of attraction" to guarantee the globally optimal solution; hence this provides the exact latent representation of objects.

For $m = 1$, the SVD solution is known for its effectiveness as a nice initialization to low-rank decomposition problems and iterate within the basin of attraction under certain conditions (Tu et al., 2015; Sun and Luo, 2015; Lu et al., 2014a; Sanghavi et al., 2017).

Lemma 3.1. Let $X \in \mathbb{R}^{n_1 \times n_2}$ be a (rectangular) low-rank matrix which has the rank-r solution $B \in \mathbb{R}^{n_1 \times r}$ and $A \in \mathbb{R}^{n_2 \times r}$ such that $X = BA^T$. Also, suppose X_0 is rank-r approximation of X (see Eq. (2.1) in (Tu et al., 2015)), and its SVD solution is $X_0 = U_0\Sigma_0V_0^T$. Then, for our goal of recovering those factors B and A, initializations $B_0 \leftarrow U_0\Sigma_0^{1/2}$ and $A_0 \leftarrow V_0\Sigma_0^{1/2}$ will be within the basin of attraction.

Proof. Specifically, let the rank-r approximation X_0 be the iterative projection result of (2.1) in (Tu et al., 2015) after $T_0 \geqslant 3\log(\sqrt{r}\kappa) + 5$ iterations where κ and r are the condition number and the rank of the original X respectively. Then, Theorem 3.3 in (Tu et al., 2015) states that the distance between B_0 and A_0 the target B and A is bounded as

$$\operatorname{dist}\left(\begin{bmatrix} B_0 \\ A_0 \end{bmatrix}, \begin{bmatrix} B \\ A \end{bmatrix}\right) \leqslant \frac{1}{4}\sigma_r(B)$$

where $\sigma_1(B) \geqslant \sigma_2(B) \geqslant \cdots \geqslant \sigma_r(B) > 0$. Also, starting from the initialization $\tau = 0$, the subsequently updated results B_τ and A_τ for $\tau = 1, 2, \ldots$ with a constant step size $\mu = 2/187$ will satisfy

$$\operatorname{dist}\left(\begin{bmatrix} B_\tau \\ A_\tau \end{bmatrix}, \begin{bmatrix} B \\ A \end{bmatrix}\right) \leqslant \frac{1}{4}\left(1 - \frac{4}{25}\frac{\mu}{\kappa}\right)^{\tau/2} \sigma_r(B)$$

for the τ-th iterations, so the subsequent solutions will only get closer to the target. Thus, the initialized $B_0 \leftarrow U_0\Sigma_0^{1/2}$ and $A_0 \leftarrow V_0\Sigma_0^{1/2}$ from the SVD solution $X_0 = U_0\Sigma_0V_0^\top$ are within the "basin of attraction". □

This lemma (based on Theorem 3.3 in (Tu et al., 2015)) can directly be applied to our relational tensor with $m = 1$ (i.e., a single "slice" X_1) in the context of relational learning which implies there is only one predicate to consider.

But our case is $m > 1$, so we perform a simple heuristic of averaging "slices" of X_k for $k = 1, \ldots, m$ to construct $\tilde{X} = \frac{1}{m}\sum_{k=1}^{m} X_k$. Then, on this new single "slice" $\tilde{X}$, we want to see if we can still initialize A_0 as described in Lemma 3.1 via the SVD solution of $\tilde{X}$ so that A_0 is still in the basin of attraction. In turns out that we can still use the SVD initialization of $\tilde{X}$ to initialize A_0 if we have m slices where m (i.e., # of possible predicates) is logarithmically dependent on n (i.e., # of object categories). Below, we show the sample complexity of m which allows us to use the SVD solution

of $\tilde{X}$ which provably puts us in the basin of attraction with an accurate SVD estimation of the latent representations. The basic idea is that, if we successfully estimate the mean of the samples (i.e., $\mathbf{E}(X)$), then a simple SVD will give us the representation of the objects. Hence our problem reduces to computing the sample complexity of estimating the mean of the distribution with respect to m and n.

Lemma 3.2. Let $\mathbf{E}(X)$ be the true abstract object relationship matrix from which X_k's are sampled from, $\epsilon > 0$ be the error of our estimate and $\delta > 0$ be the failure probability. Furthermore, assume that each X_k for $k \in \{1, \ldots, m\}$ is an independent Bernoulli random matrix. Then A is an (ϵ, δ) solution if $m = \mathcal{O}\left(\frac{1}{\epsilon} \log\left(\frac{n}{\delta}\right)\right)$.

Proof. Let $\mathbf{E}(X) = \hat{X}$ be the mean of the distribution. Since we assume that the mean can be estimated by drawing independent predicates, we can directly apply the matrix concentration inequalities that were recently developed (Tropp, 2015) which only requires that the first and second moment of the random matrix be bounded. This follows from the fact that the matrices are binary in our case. We will use $\tilde{\mathcal{O}}$ to hide log factors. First note that by triangle inequality,

$$\|X - \hat{X}\| \leqslant \|X\| + \|\hat{X}\| \tag{3.6}$$

The spectral norm of $\|X\|$ is always bounded since they are binary. Moreover, with high probability we have that, $\|X\| = \tilde{\mathcal{O}}(\sqrt{n})$ for any Bernoulli matrix (Vu, 2008) and hence the expectation has spectral norm of the same order. Hence we have that,

$$\|X - \hat{X}\| \leqslant \tilde{\mathcal{O}}(\sqrt{n}) \tag{3.7}$$

Let $\tilde{X} = \frac{1}{m}\sum_k X_k$. Using corollary 6.2.1 from (Tropp, 2015), we have that,

$$\mathbb{P}\left[\|\tilde{X} - \hat{X}\| \leqslant \epsilon\right] \leqslant \tilde{O}(n \exp(-m\epsilon)) \tag{3.8}$$

Now setting the right hand side to δ and solving for m, we get that

$$m = \tilde{\mathcal{O}}\left(\frac{1}{\epsilon}\log\frac{n}{\delta}\right) \tag{3.9}$$

as desired. Thus, the number of slices we need to obtain such SVD has a logarithmic dependence on n. □

Having initialized A, we simply set $B_k = \bar{X}A(A^TA)^{-1}$ as the initial point. Another option is to use the least squares solution $B_k = X_kA(A^TA)^{-1}$ with respect to each X_k, but this has a higher chance to overfit the data. Finally, each $R_k \in \mathbb{R}^{r\times r}$ for $k \in \{1, \ldots, m\}$ can be solved with its respective X_k given the original factorization setup Eq. (3.2):

$$R_k = (A^TA)^{-1}(A^TX_kA)(A^TA)^{-1}. \tag{3.10}$$

Alternating Block Coordinate Descent

Let us first consider problem Eq. (3.4). We see that Eq. (3.4) has multiple global optimal solutions since the value of the loss is invariant to a basis transformation: $B' = BP$ and $A' = AP^{-T}$ for any invertible matrix $P \in \mathbb{R}^{r\times r}$ has the same objective function value as B and A. Thus, we add a term that restricts such degenerate cases:

$$\lambda_p \sum_{k=1}^{m} \|A^TA - B_k^TB_k\|_F^2 \tag{3.11}$$

where $\lambda_p > 0$.

A high value of λ_p, makes the two factors B_k and A to be on the unit

"scale", or in other words, acts as a way to put A and B on "equal footing", and it serves a similar purpose for removing scale invariance of A and R which is of interest (Tu et al., 2015).

Our final model which adds the regularization in Eq. (3.11) to a formulation equivalent to Eq. (3.3) is

$$\begin{aligned}\min_{A,R_k,B_k} \sum_{k=1}^{m} \|X_k - B_k A^T\|_F^2 + \gamma \sum_{k=1}^{m} \|B_k - AR_k\|_F^2 \\ + \lambda_p \sum_{k=1}^{m} \|A^T A - B_k^T B_k\|_F^2.\end{aligned} \tag{3.12}$$

Equivalence means that there exists some $\gamma > 0$ such that the optimal solutions of Eq. (3.3) and Eq. (3.12) coincide, a direct consequence of Lagrange multiplier theory (Bertsekas, 1999). Note that the dual variable γ controls the fit to the constraint $B_k = AR_k$, so we will apply a continuation technique to solve Eq. (3.12) (without (3.11) for now) for increasing γ to enforce $B_k = AR_k$ (Nocedal and Wright, 2006). Then, we fix γ and add Eq. (3.11) to solve Eq. (3.12). We used $\lambda_p = 0.01$.

Solving for a Fixed γ. We iteratively solve for A and each B_k for $k \in \{1, \ldots, m\}$ individually with gradient descent methods as follows at each iteration. First, to solve A, we fix B_k for $k \in \{1, \ldots, m\}$ and perform gradient descent with respect to A as in line 12 of Alg. 1. Second, to solve each B_k, we fix A and $B_{\bar{k}}$ for $\bar{k} \neq k$ and perform gradient descent with respect to B_k as in line 14 of Alg. 1. To solve both of these subproblems, we used Minfunc/Schmidt solver with backtracking line search.

Note that we can solve each R_k for $k \in \{1, \ldots, m\}$ in a closed form $R_k = \left(A^T A\right)^{-1} \left(A^T B_k\right)$ since the last term does not involve any R_k (line 15 of Alg. 1). The optimization problem to solve for B_k and R_k is decomposable, so one main advantage is that they can be solved in parallel. The above procedure produces a monotonically decreasing sequence of iterates thus guaranteeing convergence (Gorski et al., 2007).

Scale Invariance

We note that this regularizer Eq. (3.11) also removes the scale invariance when $A = cA$ and $R = \frac{1}{c^2}R$ for some $c > 0$. First, recall that in our basic objective function without the regularizer Eq. (3.2), ARA^T is invariant to scaling $A' = cA$ and $R' = \frac{1}{c^2}R$ for some $c > 0$ because

$$A'R'(A')^T = (cA)\frac{1}{c^2}R(cA^T) = ARA^T.$$

which is solved as in Eq. (3.3). We *want* to restrict the invariance from such scaling, and this regularizer achieves this scaling invariance.

Lemma 3.3. The regularizer $\|A^TA - B^TB\|_F^2$ is variant to scaling A and R where the constraint $B = AR$ is imposed as in Eq. (3.3).

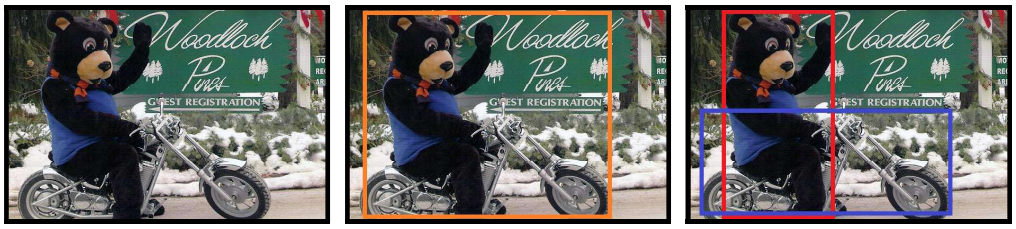
Proof. Suppose we scale A and R such that $A' = cA$ and $R' = \frac{1}{c^2}R$ for some $c > 0$ (except for $c = 1$ which we do not consider as scaling), and we similarly let $B' = A'R'$ be the constraint corresponding to A' and R'. The regularizer is variant to such scaling when $\|(A')^TA' - (B')^TB'\|_F^2 \neq \|A^TA - B^TB\|_F^2$. This holds when $(A')^TA' - (B')^TB' \neq A^TA - B^TB$, and we see that

$$(A')^TA' - (B')^TB' = c^2A^TA - \frac{1}{c^4}R^TA^TAR^T \neq A^TA - R^TA^TAR = A^TA - B^TB$$

hence $\|A^TA - B^TB\|_F^2$ is variant to scaling. □

Then, since the regularizer is variant to scaling based on Lemma 3.3, the new objective function which now includes the regularizer to Eq. (3.12) is also variant to scaling since

$$\|X - B'(A')^T\|_F^2 + \|(A')^TA' - (B')^TB'\|_F^2 \neq \|X - BA^T\|_F^2 + \|A^TA - B^TB\|_F^2.$$



(a) Predicate (b) Phrase (c) Relationship

Figure 3.4: **Detection task conditions.** Given object bounding boxes: (a) Predicate (easy): does not require bounding boxes. (b) Phrase (moderate): requires relationship bounding box (orange) containing both objects. (c) Relationship (hard): requires individual bounding boxes (red/blue).

even if $A'R'(A')^T = ARA^T$ is scale invariant. This regularizer from (Tu et al., 2015) originally acts as a way to put A and B on "equal footing", and it serves a similar purpose for removing scale invariance of A and R which is of interest.

Scene Graph Prediction Pipeline

We now describe the training procedure of the pipeline (Fig. 3.2).

Relational Learning Module. We first setup the RL module by constructing the multi-relational tensor $X \in \mathbb{R}^{n \times n \times m}$ on the Visual Genome dataset as described before. Then, for $r = 15$, we solve for the latent representation of the objects $A \in \mathbb{R}^{n \times r}$ and the factor matrices $R_1, \ldots, R_m \in \mathbb{R}^{r \times r}$ based on Eq. (3.12) as in Alg. 1. Next, using the trained A and $R_1, \ldots, R_m$, we reconstruct the low-rank multi-relational matrix $\hat{X}$ which is the stack of m low-rank relational matrices similar to X except that each slice is $\hat{X}_k = AR_kA^T$ for $k \in \{1, \ldots, m\}$. Then, given objects i and j, the predicted predicate distribution is $k_{RL} = softmax(\hat{X}(i, j, :)) \in \mathbb{R}^m$.

Training the Pipeline. Given an image, the initial object bounding boxes are detected via a Region Proposal Network (Ren et al., 2015) to train our end-to-end pipeline for scene graph prediction which consists of two modules (see Fig. 3.2). (a) **Scene graph (SG) module**: The iterative message passing network by (Xu et al., 2017) predicts both objects and predicates

concurrently. (b) **Relational learning (RL) module**: Our tensor-based relational learning provides predicate prior $\hat{X}(i, j, :)$ between object i and j where the low-rank tensor is now constructed based on the entire Visual Genome training set. Given object-subject bounding boxes, our pipeline trains its relationship as follows: (1) The SG module estimates the object labels i^* and j^* along with the predicate distribution $k^*_{SG} \in \mathbb{R}^m$. (2) The RL module computes the predicate prior based on those estimates: $k^*_{RL} = softmax(\hat{X}(i^*, j^*, :)) \in \mathbb{R}^m$. (3) The prior k^*_{RL} is stochastically applied to the network-based estimate k^*_{SG} as

$$k^* = k^*_{SG} \odot D(k^*_{RL}, \theta)$$

where $\odot$ is the Hadamard product and $D(y, \theta) \in \mathbb{R}^m$ is a 'y-or-1' filter where the i'th element is $y(i)$ with probability θ or 1 with probability $1 - \theta$. This stochastic process balances the influence of the prior k^*_{RL} while effectively injecting 'global' predicate prior which regularizes under- or over-estimated predicates during the training. (4) Using the new predicate prediction k^*, relationship loss is now computed and backpropagated to the SG module with respect to both objects *and* predicates. In order to avoid the SG module prematurely relying too much on the preconstructed RL module at early iterations, we first lightly train the SG module without the RL module ($\theta = 0$) and then include the RL module ($\theta = 0.2$) for further iterations.

3.5 Experiments

We evaluate our model on two datasets. First, we test our regularization model as a standalone method on the Scene Graph dataset (Johnson et al., 2015) and compare against the relationship detection method by Lu et al. (Lu et al., 2016). To show that performance gains are *not* just from the decomposition formulation ((3.1)), we also compare against Tucker 2

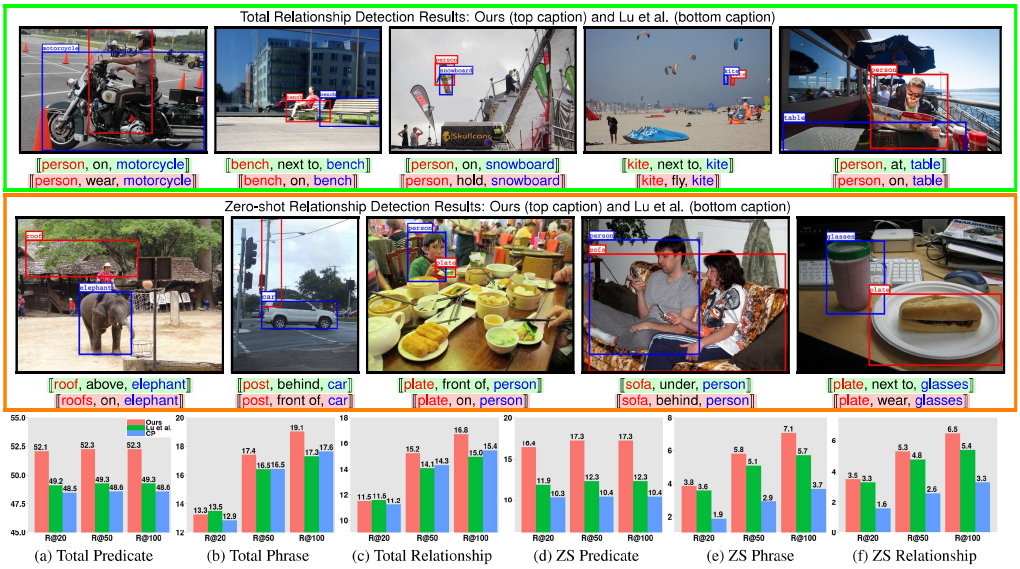


Figure 3.5: The total visual relationship detection (top row in green box) and the zero-shot visual relationship detection results (middle row in orange box) on Scene Graph dataset using our algorithm (top caption) and (Lu et al., 2016) (bottom caption). The correct and incorrect predictions are highlighted in green and red respectively. Visual relationship detection results (bottom row) on Scene Graph using ours (**red**), Lu et al. (**green**) and CP (**blue**).

(Tucker, 1966) and PARAFAC (Harshman and Lundy, 1994). Second, for more difficult scene graph prediction tasks on Visual Genome (Krishna et al., 2016), we show significant improvements over the recent state-of-the-art message passing network model by Xu et al. (Xu et al., 2017) using our end-to-end pipeline that integrates our tensor-based relational module with their message passing model (Xu et al., 2017). The dense prior inferred from our provably robust relational module directly influences both the training and testing of the pipeline in a holistic manner as shown in Fig. 3.2. In both evaluations, we measure the true positive rate from the top p confident predictions referred to as recall at p (R@p) since not all ground truth labels can be annotated.

Scene Graph Dataset

We used the same set of 5000 training (<1% unique tuples) and 1000 test images with $n = 100$ object categories and $m = 70$ predicates as in (Lu et al., 2016).

Visual Relationship Prediction Setup. The procedure of constructing the low-rank multi-relational matrix $\hat{X}$ is identical to the previous description where in this case we use the Scene Graph dataset. Then, the predicted predicate between object i and j is $k^* = \text{argmax}_k \, \phi_{ij} \hat{X}(i, j, k)$ based on a vector of 'probability distribution' of predicates.

Prediction Tasks. We setup three different prediction experiments of varying difficulties (see Fig. 3.4): **(a)** Predicate, **(b)** Phrase and **(c)** Relationship predictions. These are performed at R@p for $p \in \{100, 50, 20\}$ in two settings: **(1)** Total and **(2)** Zero-shot (test set *not* observed in training).

Tucker 2 Results

Below, we show the experiment results on the Scene Graph dataset using Tucker 2-decomposition (Tucker, 1966).

	Total			**Zero-shot**		
	R@100	R@50	R@20	R@100	R@50	R@20
Predicate	11.86	11.86	10.81	2.57	2.57	2.57
Phrase	3.74	2.97	2.07	0.68	0.43	0.17
Relationship	3.29	2.64	1.82	0.60	0.43	0.17

Table 3.1: Results on Scene Graph using Tucker 2 (Tucker, 1966).

Note that all the results are very underwhelming compared to the results of the other methods potentially due to the large degrees of freedom that Tucker 2 innately possesses which may lead to weak generalization and prediction power.

Simple Baseline Result

We also tested the following naive baseline method which can naively predict the unobserved relationships (e.g., [[obj1, ?, obj2]]).

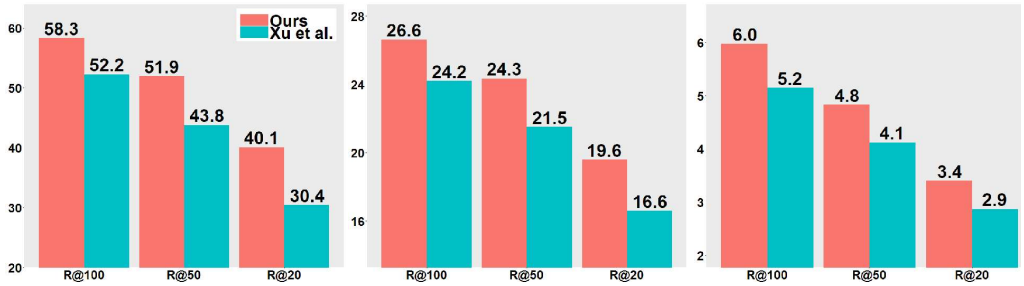
1. For obj1, find the predicate with the maximum number of occurrence (if tied, pick one randomly) regardless of obj2.
2. Similarly, for obj2, find the predicate with the maximum number of occurrence (if tied, pick one randomly) regardless of obj1.
3. Then, from the two best predicates with respect to each of obj1 and obj2, choose the one with the higher number of occurrence (if tied, pick one randomly). This will be the final predicate for [[obj1, ?, obj2]].

This essentially picks the most occurring predicate that considers either obj1 or obj2 (e.g., "given person for obj1, choose the most occurring predicate that follows person without considering the corresponding obj1"). We call this the "Back-off Baseline" for its simplicity. The results are shown in Table 3.2.

	ZS Predicate			ZS Phrase			ZS Relationship		
Methods	R@20	R@50	R@100	R@20	R@50	R@100	R@20	R@50	R@100
Ours	16.4	17.3	17.3	3.8	5.8	7.1	3.5	5.3	6.5
Lu et al.	11.9	12.3	12.3	3.6	5.1	5.7	3.3	4.8	5.4
PARAFAC	10.3	10.4	10.4	1.9	2.9	3.7	1.6	2.6	3.3
Back-off	9.3	9.4	9.4	1.2	1.5	1.6	1.0	1.3	1.5

Table 3.2: Zero-shot results for Scene Graph dataset experiment

We observed that the performance of the Back-off Baseline approach was relatively worse in all setups, but the predicate prediction results (ZS Predicate, left 3 columns) were not that much worse than the PARAFAC decomposition. This implies that the raw relationship occurrence counts from the training data may already provide a reasonable amount of information so that the above heuristic serves as a nice baseline, at least for the current benchmark.



(a) PredCls (b) SgCls (c) SgGen

Figure 3.6: Scene graph detection task (see Table 3.3) results on Visual Genome using ours (**red**) and (Xu et al., 2017) (**cyan**). Our pipeline *without* the RL module show results similar to (Xu et al., 2017) (**cyan**).

Visual Genome Dataset

We used the cleaned up version of the dataset following (Xu et al., 2017) to account for poor/ambiguous annotations which consists of 108,077 images, 25 objects, and 22 relationships where we used 70% for training and 30% for testing. For the experiments, we used the most appearing $n = 150$ object categories and $m = 50$ predicates (11.5 objects and 6.2 relationships per image on average).

Prediction Setup. Once the pipeline is trained, the prediction result is simply the forward propagation output of the pipeline except we now set $\theta = 1$ to fully use the relational prior k^*_{RL}.

Scene Graph Prediction Tasks. Detecting a scene graph requires inference on three parts: predicate, object class and bounding box which requires accurate predictions on these parts incrementally (Xu et al., 2017) as shown in Table 3.3. For all these tasks, we used R@p for $p \in \{100, 50, 20\}$.

Results on Relationship Learning Tasks

Visual Relationship Detection on Scene Graph. We show visual relationship detection results on the Scene Graph dataset using CP (Harshman and Lundy, 1994), Lu et al. (Lu et al., 2016) and our algorithm at the bottom of Fig. 3.5. For all tasks, our results outperform other methods. Especially,

Prediction Tasks	Predicate	Object	B-box
Predict Predicate (**PredCls**)	✓		
Classify SG (**SgCls**)	✓	✓	
Generate SG (**SgGen**)	✓	✓	✓

Table 3.3: Scene Graph detection tasks. Check marks indicate required prediction components. The tasks become incrementally more demanding from top (PredCls) to bottom (SgGen).

our zero-shot prediction (Fig. 3.5 (d)) results substantially outperform the state-of-the-art ((Lu et al., 2016)) by ~ 40% in all recalls. In much more difficult phrase (b,e) and relationship (c,f) detection (Fig. 3.5), we achieve improvements in all tasks under almost all recalls. We observe that our *zero-shot* predicate detection results (Fig. 3.5 (d)) given *known* object pairs is competitive with the *total* phrase detection results by (Lu et al., 2016) (Fig. 3.5 (b)) given *unknown* object pairs. This implies that while accurate object detection is crucial for visual relationship detection, more difficult zero-shot learning is a *less* critical factor for our algorithm.

Scene Graph Prediction on Visual Genome. We now show the scene graph prediction results (Fig. 3.6) on Visual Genome using Xu et al. (Xu et al., 2017) and our pipeline (Fig. 3.2). We also evaluated (Lu et al., 2016) on the same tasks, but the model did not scale well to the task complexity so the performances were lower than the other two methods by large margins. **(a)** PredCls: Our model provides significant improvements in the predicate detection tasks in all recalls by at most ~30% in R@20. Since this task only demands predicate predictions, such large improvements demonstrate that the tensor-based RL module functions as an effective prior for inferring visual relationships by better utilizing the large but sparse dataset. **(b)** SgCls: The results on the scene graph classification (Fig. 3.6(b)) show that our model improves object classifications as well in all recalls where our R@50 result is on par with R@100 of (Xu et al., 2017). The boost in predicate prediction improves overall inference on the interconnected object and predicate inference of the SG module (Xu et al.,

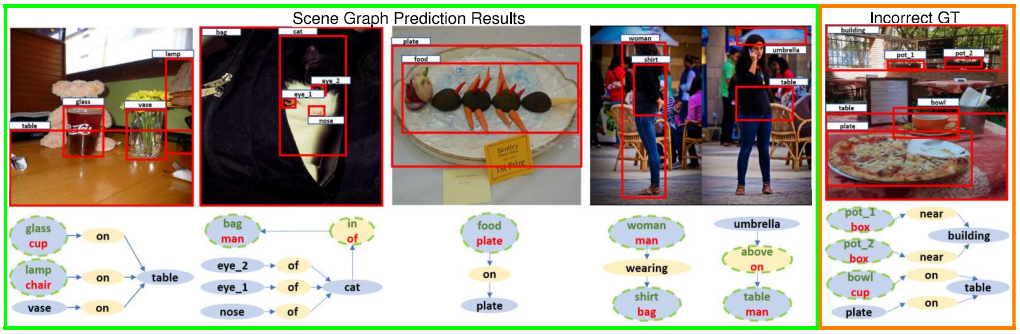


Figure 3.7: Scene graph classification results on Visual Genome using ours and (Xu et al., 2017). For each column, the predicted objects (blue ellipses) and their relationships (yellow ellipses) are constructed as a scene graph its top image. The bounding boxes labels reflect our prediction results. For difficult predictions (**green dashed boundary**) where our model has correctly predicted (top **green**) and while (Xu et al., 2017) has misclassified (bottom **red**) are shown. The rightmost column is an example of a case where our model provides more accurate predictions (**pot** and **bowl**) than those of the ground truth (**box** and **cup**).

2017) during the training. **(c)** SgGen: On the last task which also predicts the bounding box, our model showed ~10% improvements in all recalls over (Xu et al., 2017).

Remarks. We observe that our RL module provides boosts on not only the predicate detection (PredCls) but also the interdependent object classification tasks (SgCls and SgGen) enabled by our composite pipeline (Fig. 3.2), and this is our initial hypothesis: relationship learning is a bottleneck which needs to be focused on. Second, as seen in the rightmost column of Fig. 3.7, such rare mislabeled or semantically ambiguous samples become extremely difficult to infer, but the prior from the RL module could provide strong 'advice' on such outliers based from its dense knowledge spanning *entire* relationship space.

3.6 Summary

We presented a novel end-to-end pipeline for the visual relationship detection problem. We first exploits a simple tensorial representation of the

training data and derives a powerful relational prior based on a algebraic formulation to obtain latent "factorial" representations from the sparse tensor via a novel spectral initialization. Our results suggest that the factors can be provably learned from observations only logarithmic in the number of relationships given the ill-posedness of the problem. With this regularization, we show how informing an end-to-end visual relationship detection pipeline with such a distilled prior constructed from the *latent representations of objects and their relationships* yields state-of-the-art in various experiments from predicate to scene graph prediction.

4 COUPLING HARMONIC BASES FOR CROSS-SECTIONAL AND LONGITUDINAL CHARACTERIZATION OF BRAIN CONNECTIVITY EVOLUTION

The visual relationships described in the previous chapter demonstrated a cross-sectional (i.e., non-temporal) relationship between the objects in images. Starting from this chapter, we begin introducing another common type of relationship, the *temporal relationship*, containing the sequential information of the data. In this chapter specifically, we will derive the latent representation of *brain networks* while imposing the cross-sectional and temporal consistency directly in the latent space (see Fig. 1.6).

4.1 Overview

Large scale scientific initiatives such as the Human Connectome Project (HCP) are beginning to provide exquisite imaging data that may eventually enable a full structural connectivity mapping of the human brain (Van Essen et al., 2012). For instance, diffusion magnetic resonance imaging (dMRI), an imaging modality central to the aforementioned studies, captures in a spatially localized (voxel-wise) manner, water diffusion properties that can be used to infer the arrangement of network pathways in the brain (Sotiropoulos et al., 2013; Uğurbil et al., 2013). After suitable pre-processing, e.g., via so-called tractography procedures, we obtain a unique view of the fiber bundle layout that connects distinct brain regions (Jbabdi et al., 2015) (see Fig. 4.1). From an analysis perspective, once the spatial organization of these fiber bundles is expressed as a graph whose nodes represent separate brain regions and the edges denote the "strength" of connection (e.g., number) of the connecting inter-region fibers, a variety of analyses can be conducted (Kim et al., 2013, 2015; Sporns, 2011; Wig

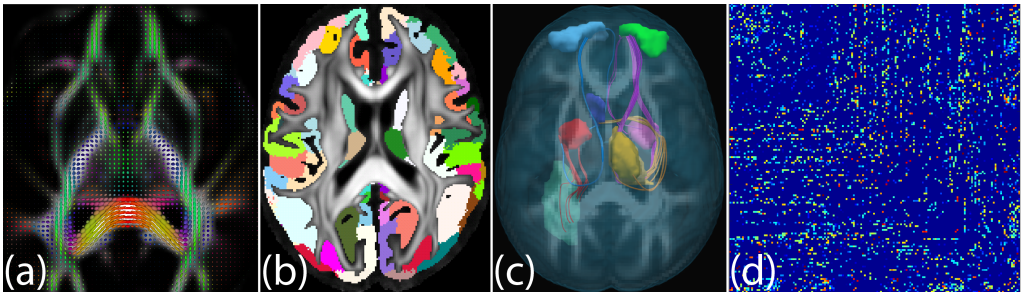


Figure 4.1: Deriving connectivity matrix from dMRI. (a) Diffusion tensor ellipsoids obtained from the dMRI data using non-linear estimation. (b) Anatomical regions in cortical and sub-cortical gray matter are used to define the nodes in the brain network. (c) Fiber tracts (axonal pathways between brain regions) estimated via tractography are used to define connectivity strength between various gray matter nodes in the brain. (d) The brain networks can be represented as symmetric adjacency matrices.

et al., 2011). For example, we may ask whether a specific *edge* of the graph exhibits statistically different connectivity measurements across clinically disparate groups: diseased and healthy. If the results of this hypothesis test are statistically significant, we can conclude that the corresponding fiber bundle (pertaining to the edge) is possibly affected by the disease.

But more recently, there is increasing interest in identifying not just imaging based or structural connectivity based biomarkers, rather to quantitatively characterize disease *progression* (Cairns et al., 2015; Raj et al., 2012; Wang et al., 2015). For example, studies may recruit subjects for multiple visits over a period of time (which varies from months to years) and acquire diffusion imaging data at several time points. In such *longitudinal* datasets, each subject (or sample) corresponds to multiple images (or the corresponding brain connectivities) at different time points. The scientific goal then is to identify the entire life cycle of brain connectivity evolution — from when a middle aged participant was healthy to a stage where the individual's cognitive function has become much worse.

The standard approach to answering the question above is to characterize change in brain connectivity at the level of individual edges in the graph. There are two problems with this proposal. First, treating individ-

ual edges as primitives neglects the local context in which the edge exists in the actual object of interest — i.e., the entire connectivity graph of the individual. But from a technical perspective, a second issue is perhaps more important. For n regions, we obtain $\mathcal{O}(n^2)$ edges. After learning a model (e.g., association of the connection weights with age) for each edge and estimating the significance of the fit (e.g., p-values), we *cannot* simply report the edges with small p-values as relevant. Since the analysis is being conducted for a large number of edges, the likelihood of false positives is high, so a conservative multiple comparisons correction needs to be adopted. This correction may often be too conservative, and we may end up discarding connections that are, in fact, scientifically meaningful; this is an undesirable consequence of treating the edges individually. To avoid this problem, practitioners often rely on summary measures of the connectivity graph instead, such as clustering coefficients, small-worldness, modularity and so on (Achard et al., 2006; Rubinov and Sporns, 2010). This works well but is limited in that we cannot uncover spatially localized effects of disease (or other covariates) on connectivity.

A more attractive solution is to think of the graphs as an object and utilize a suitable parameterization of the graph. One possibility is to use the Laplacians (Reijneveld et al., 2007; Stam and Reijneveld, 2007), either at the level of individual subjects or in terms of distinct partitions of the full cohort progressively going from healthy to diseased, i.e., the first partition is comprised of completely healthy individuals whereas the k-th partition includes diseased subjects. Then, if we look at the full set of bases of the partition-specific Laplacian we can come up with ways to characterize change in these bases as we move from the healthy to the diseased partition. Of course, such a parameterization also enables *longitudinal* analysis. For example, if we have data for multiple time-points for each partition, we can track how the Laplacian bases evolve over time. To achieve these goals, i.e., for analyzing changes across partitions of dis-

ease severity or partition-specific longitudinal analysis, one requirement is the ability to derive a *coupled set of harmonic bases* for a set of ordered Laplacians (longitudinally *and* cross-sectionally). This allows treating the full data holistically *while preserving the ordinal nature of time and disease-induced cognitive decline*. While a mature body of literature in numerical analysis provides sophisticated ways of deriving orthonormal set of bases for any self-adjoint operator, it provides little guidance on how to impose the coupling requirement, essential in this application. For example, we find that for most of the widely used eigenvalue decomposition methods (Saad, 1992; Warsa et al., 2004), it is non-trivial to modify the numerical scheme to satisfy the consistency requirement between consecutive set of eigen-bases. Addressing this limitation is a goal of this chapter.

Contributions

With the foregoing motivation, the core of this work deals with deriving efficient numerical optimization schemes to solve an ordinally and longitudinally coupled set of a generalized eigenvalue problems. To our knowledge, few publicly available alternatives currently exist (Kovnatsky et al., 2013; Lei and Li, 2009). We provide **(i)** a novel formulation for estimating harmonic bases of brain connectivity networks that are smoothly varying in terms of both longitudinal as well as cross-sectional ordering (induced by a separate covariate such as cognitive performance). We provide an iterative numerical scheme for solving the problem using stochastic block coordinate descent based manifold optimization techniques. **(ii)** We show that such a framework provides an exciting scientific tool in the following sense. Once the model has been estimated, we can vary a single parameter and "see" how the structural brain connectivity of an individual evolves over time or as a function of disease (see Fig. 1.3 for a qualitative demonstration). This yields a valuable mechanism for performing individual-level prediction. In fact, in the following chapters (Chapter 5

and Chapter 6), we further develop such individual-level predictions not only for the brain connectivity but also for the ROI-based measures in the context of AD (see Fig. 7.3 for an example of ROI-specific pathology trajectory prediction). We show how our algorithm is able to provide connectivity prediction in a population of *healthy controls* who have some known risk factors of AD. Even though these individuals are *asymptomatic*, our approach is able to obtain a nominal degree of accuracy in assigning the subject to distinct cognitive quantiles. This demonstrates that we can, in fact, obtain a better than chance accuracy where the disease signal is so weak which is the main contribution of this work.

4.2 Coupled harmonic bases for brain networks

Parameterization of Brain Network

Let us first describe a simple procedure for parameterizing the brain connectivity network in terms of its bases for individual subjects. Let A be a $n \times n$ weighted adjacency graph, A as in Fig. 4.1(d) representing a brain connectivity between n regions of the brain for a subject. We construct the Laplacian L, a commonly used tool/parameterization for representing graphs, defined as

$$L = D - A, \quad D(i,i) = \sum_{j=1}^{n} A(i,j),$$

where D is called the degree matrix. Based on spectral graph theory, the eigenvectors corresponding to lower order eigenvalues contain the 'low frequency' information which reflects the latent structure of the Laplacian. The bases are estimated by minimizing the following objective function

$$\min_{V \in \mathbb{R}^{n \times p}} \operatorname{tr}(V^T L V), \quad \text{s.t.} \quad V^T V = I, \tag{4.1}$$

where $\mathrm{tr}(\cdot)$ is the trace functional. The solution V to the above numerical optimization problem consists of the eigenvectors associated with the p smallest eigenvalues of L which we solve to express a given brain connectivity network via its Laplacian and/or its p eigenvectors/eigenvalues.

Now suppose we are given a longitudinal dMRI dataset for N subjects with T time points: this provides NT brain networks. We can parameterize all the networks simultaneously by minimizing the following objective function

$$\begin{aligned} \min_{V_{[i,j]} \in \mathbb{R}^{n \times p}} \; & \sum_{i=1}^{N} \sum_{j=1}^{T} \mathrm{tr}(V_{[i,j]}^{T} L_{[i,j]} V_{[i,j]}) \\ \text{s.t.} \; & V_{[i,j]}^{T} V_{[i,j]} = I, \end{aligned} \tag{4.2}$$

where $L_{[i,j]}$ denotes the Laplacian matrix of brain network for subject i and time point j and $V_{[i,j]}$ denotes the set of p eigenvectors for $L_{[i,j]}$.

However, this formulation ignores a couple of key properties of our analysis goal (conceptually shown in Fig. 4.2). (1) Each subject has multiple time points which means that not all networks in the population are 'independent'. There are strong dependencies among the networks derived from a single person observed over time. (2) The subjects can, if desired, be partitioned into distinct groups if a covariate of interest for the subjects is close enough (i.e., similar cognitive scores or a measure of pathology such as amyloid protein load may have roughly similar connectivity strength (Drzezga et al., 2011)). This suggests that the bases that we find must also be related, or *coupled*, while still respecting, to the extent possible, their original Laplacians. If we consider the population of networks as a system, the recovery of the full set of bases must ensure a notion of consistency among $\{V_{[i,j]}\}$, governed by either the cognitive score grouping or longitudinal ordering described above.

We now present our proposed framework for adding constraints in Eq. (4.2) that will ensure that the full set of Laplacians are treated jointly. Without loss of generality, we can work with the example dataset scenario

presented in Fig. 4.2.

Longitudinal Coupling

In this section, we introduce basis coupling constraints that model the relationships (blue arrows in Fig. 4.2) between *temporally* consecutive bases. Suppose we consider the bases $V_{[\bullet,j]}$ and $V_{[\bullet,j+1]}$ for two consecutive time points j and $j+1$ for a specific subject. Since these are derived from the Laplacians of the same subject, we impose a homology constraint between the latent structures. In other words, we expect that $V_{[\bullet,j]}$ and $V_{[\bullet,j+1]}$ differ only by a small degree of *rotation*. Specifically, we impose

$$V_{[\bullet,j+1]} = R^{\bullet}_{[j,j+1]} V_{[\bullet,j]}, \tag{4.3}$$

where $R^{\bullet}_{[j,j+1]} \in \mathbb{SO}(n)$ which is a group of $n \times n$ orthogonal matrices with determinant $= +1$. This is, in fact, a Procrustes problem (Wang and Mahadevan, 2008) of aligning the bases which provides the longitudinal evolution process of the set of bases as a sequence of rotation matrices. Note that the rotation matrix which aligns $V_{[\bullet,j+1]}$ to $V_{[\bullet,j]}$ is simply $R^{\bullet}_{[j+1,j]} = R^{\bullet^T}_{[j,j+1]}$.

Now, for eigenvectors $V_{[\bullet,j]}$ and $V_{[\bullet,j+1]}$, we see that

$$V^T_{[\bullet,j]} V_{[\bullet,j]} = V^T_{[\bullet,j]} R^{\bullet}_{[j+1,j]} V_{[\bullet,j+1]} = I, \tag{4.4}$$

$$V^T_{[\bullet,j+1]} V_{[\bullet,j+1]} = V^T_{[\bullet,j+1]} R^{\bullet}_{[j,j+1]} V_{[\bullet,j]} = I. \tag{4.5}$$

Multiplying the above two equations we have

$$\begin{aligned} &\left(V^T_{[\bullet,j]} R^{\bullet}_{[j+1,j]} V_{[\bullet,j+1]}\right)\left(V^T_{[\bullet,j+1]} R^{\bullet}_{[j,j+1]} V_{[\bullet,j]}\right) = I \\ \implies &V^T_{[\bullet,j]} \underbrace{\left(R^{\bullet}_{[j+1,j]} V_{[\bullet,j+1]} V^T_{[\bullet,j+1]} R^{\bullet^T}_{[j+1,j]}\right)}_{M'_{[j+1,j]}} V_{[\bullet,j]} = I. \end{aligned}$$

Thus, we have now added a constraint on $V_{[\bullet,j]}$ that addresses the coupling between j and $j+1$ as

$$V^T_{[\bullet,j]} M'_{[j+1,j]} V_{[\bullet,j]} = I. \tag{4.6}$$

Note that for $j \notin \{1, T\}$, each $V_{[\bullet,j]}$ is tied to $V_{[\bullet,j-1]}$ and $V_{[\bullet,j+1]}$. Thus, the coupling matrices for those two relationships are $M'_{[j-1,j]}$ and $M'_{[j+1,j]}$ respectively. To account for both relations, we take the average coupling matrix as (Ham et al., 2005):

$$M_{[\bullet,j]} = \frac{M'_{[j-1,j]} + M'_{[j+1,j]}}{2}. \tag{4.7}$$

For the boundary values of $j = 1$ and $j = T$, $M_{[\bullet,1]} = M'_{[2,1]}$ and $M_{[\bullet,T]} = M'_{[T,T-1]}$ respectively. Thus, for a subject i, we derive the following optimization model for discovering longitudinally coupled bases for the brain connectivity Laplacians,

$$\min_{V_{[i,j]} \in \mathbb{R}^{n \times p}} \; \mathrm{tr}(V_{[i,j]}^T L_{[i,j]} V_{[i,j]}), \;\; \text{s.t.} \;\; V_{[i,j]}^T M_{[i,j]} V_{[i,j]} = I. \tag{4.8}$$

Recall that the above constraint is the so-called generalized Stiefel constraint (Absil et al., 2009) with the *mass matrix* $M_{[i,j]}$. This is also known as the generalized eigenvalue problem that finds the first p smallest eigenvalues and the respective eigenvectors of $L_{[i,j]}$ for the mass matrix $M_{[i,j]}$. We can finally extend this implementation for all subjects as follows,

$$\begin{aligned} \min_{V_{[i,j]} \in \mathbb{R}^{n \times p}} \;\; & \sum_{i=1}^{N} \sum_{j=1}^{T} \mathrm{tr}(V_{[i,j]}^T L_{[i,j]} V_{[i,j]}) \\ \text{s.t.} \;\; & V_{[i,j]}^T M_{[i,j]} V_{[i,j]} = I. \end{aligned} \tag{4.9}$$

We point out that for each i and j, the above model is equivalent to Eq. (4.8).

Cross-sectional Coupling

In this section, we present the appropriate constraints that will encode cross-sectional dependencies among the eigenvectors $\{V_{[i,j]}\}$. Let us say the population can be partitioned into K distinct groups (columns in Fig. 4.2) where each group/column is cognitively equivalent based on some battery of tests. Such partitions may also be derived by certain measures of pathology. For each group, we first construct an average

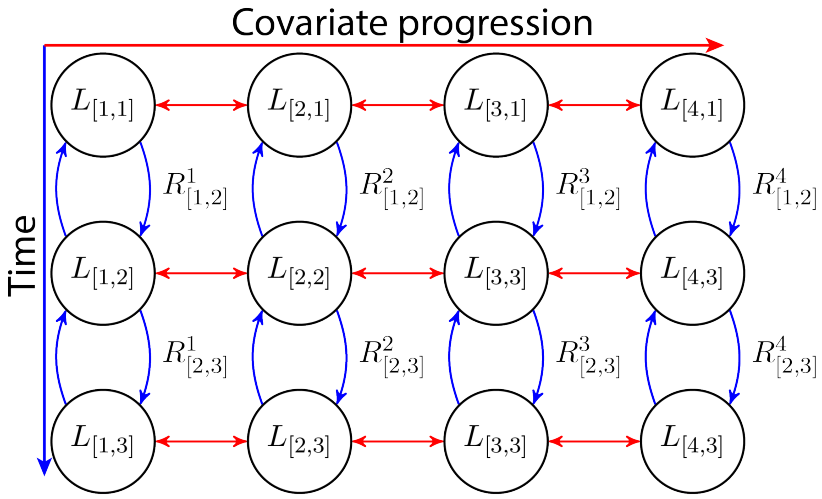


Figure 4.2: The graph representation of the coupled data matrices. The nodes in each row (cross-sectional) are coupled horizontally in red while the nodes in each column (longitudinal) are coupled vertically in blue.

Laplacian $X_{[i,\bullet]}$ at a fixed time point. This average Laplacian serves as a representative of that specific group. In principle, we can work with individual level Laplacians L, however, since the goal is to formulate the coupling of bases with respect to the covariates, the averaging helps us reduce the individual level variability and provides a more succinct picture of the network evolution along the trajectory of that covariate (e.g., cognitive scores).

Let us consider three such average Laplacians $X_{[i-1,\bullet]}$, $X_{[i,\bullet]}$ and $X_{[i+1,\bullet]}$ from three consecutive/ordered partitions. The corresponding eigenvectors will be $V_{[i-1,\bullet]}$, $V_{[i,\bullet]}$ and $V_{[i+1,\bullet]}$. Since these bases are derived from partitions with disjoint/distinct groups of subjects, we *cannot* assume a homological relationship between them. Since the coupling constraints will be added only between adjacent partitions it is nonetheless reasonable to assume that the bases will not change drastically affecting the full connectivity network. We encode this requirement as a sparsity constraint on the difference of the bases, e.g., via ℓ_0 norm. We use the relaxed ℓ_1

alternative,

$$g(V_{[i,\bullet]}) = \lambda \left(\|V_{[i-1,\bullet]} - V_{[i,\bullet]}\|_1 + \|V_{[i,\bullet]} - V_{[i+1,\bullet]}\|_1 \right) \tag{4.10}$$

where $\lambda > 0$ is the regularization parameter which controls only the magnitude of the cross-sectional coupling.

Intuitively, it enforces similarities in certain dimensions of the cross-sectional bases while allowing the others to vary freely. In other words, we preserve some structural consistencies across the groups while still allowing the group-wise bases to be different. Note that for $i = 1$ and $i = K$, the regularization terms will only contain the first term and the second term of (4.10) respectively. The *cross-sectionally coupled* bases ($V_{[i,\bullet]}$) can then be estimated by minimizing the following,

$$\begin{aligned} \min_{V_{[i,\bullet]} \in \mathbb{R}^{n \times p}} \; & \operatorname{tr}(V_{[i,\bullet]}^T X_{[i,\bullet]} V_{[i,\bullet]}) + \lambda g(V_{[i,\bullet]}) \\ \text{s.t.} \quad & V_{[i,\bullet]}^T M_{[i,\bullet]} V_{[i,\bullet]} = I. \end{aligned} \tag{4.11}$$

Putting it all together, we have the following *coupled* generalized eigenvalue formulation,

$$\begin{aligned} \min_{V_{[i,j]}} \; & \sum_{i=1}^{K} \sum_{j=1}^{T} \operatorname{tr}(V_{[i,j]}^T X_{[i,j]} V_{[i,j]}) + \lambda \sum_{i=1}^{K-1} \sum_{j=1}^{T} \|V_{[i+1,j]} - V_{[i,j]}\|_1 \\ \text{s.t.} \quad & V_{[i,j]}^T M_{[i,j]} V_{[i,j]} = I; \quad V_{[i,j]} \in \mathbb{R}^{n \times p}. \end{aligned} \tag{4.12}$$

We have now imposed both the longitudinal and cross-sectional basis coupling: for all partitions and time points.

4.3 Optimization scheme for coupled bases

In this section, we present an efficient numerical procedure for solving Eq. (4.12). Recall that the constraints involving the mass matrix form the generalized Stiefel manifold. For a few relevant technical details of the harmonic basis and Stiefel manifold, we suggest the reader to refer back to Background 2.1.

Algorithm 2 Stochastic block coordinate descent in $\mathrm{GF}_{n,p}$

Given: $f : \mathrm{GF}_{n,p} \to \mathbb{R}$, $V \in \mathrm{GF}_{n,p}(M)$, $M \in \mathbb{R}^{n \times n}$
while Convergence criteria not met **do**
  $\mathcal{S}$:= Subproblem row indices
  P_0 := Initial feasible submatrix (4.19)
  G := Subdifferential of f w.r.t. P_0 (4.20)
  W := Descent curve in the direction of $-G$ on $\mathrm{GF}_{s,p}(M_{\mathcal{S}\mathcal{S}})$ at P_0 (4.22)
  τ := Step size under strong Wolfe conditions (Nocedal and Wright, 2006)
  P := Feasible point $W(\tau)$ of subproblem with sufficient decrease in f
  $V'(P)$:= Update new feasible point (4.23)
end while

Our main strategy is to perform block coordinate descent over $\mathrm{GF}_{n,p}$ to solve for $\{V_{[i,j]}\}$ for each matrix $X_{[i,j]}$ given in Eq. (4.12). Specifically, our algorithm iteratively decreases the objective value of the problem by finding the next feasible point in a curve which lies in the generalized Stiefel manifold $\mathrm{GF}_{n,p}$ described in the constraints of (4.8), by adapting the scheme in (Collins et al., 2014; Hwang et al., 2015; Wen and Yin, 2013). This process is invoked as a module within our full pipeline.

Next, we show the entire framework of our algorithm which solves for all $V_{[i,j]}$ of the model (4.12) by iteratively solving for each $V_{[i,j]}$ while fixing the other decision variables $V_{[i',j']}$, $\forall i' \neq i \& j' \neq j$. In each iteration, we also update the mass matrix $M_{[i,j]}$ using the most recent bases.

Stochastic Block Coordinate Descent in $\mathrm{GF}_{n,p}$.

For simplicity, let us focus on a single arbitrary partition and its Laplacian X, mass matrix M and the eigenvectors V and setup the following coupled

model as in Eq. (4.11):

$$\begin{aligned} \min_{V \in \mathbb{R}^{n \times p}} \; & \operatorname{tr}(V^T X V) + \lambda g(V) \\ \text{s.t.} \quad & V^T M V = I, \end{aligned} \tag{4.13}$$

where the regularization term $g(V)$ from Eq. (4.10). First, we show how to solve Eq. (4.13) on a subset of the dimensions. This is a very common procedure in a coordinate descent method where the dimensions can be computed nearly independently to allow parallelization and make large-scale implementation possible. Specifically, we construct a subproblem for each submatrix $V_{\mathcal{S}\cdot} \in \mathbb{R}^{s \times p}$ where $\mathcal{S}$ is a subset of s row indices of V. We choose this submatrix as the *free variable* which we ultimately solve for while fixing the complementary submatrix $V_{\bar{\mathcal{S}}\cdot}$ for the rows $\bar{\mathcal{S}}$ called the *fixed variable* which we essentially treat as constants. Assuming w.l.o.g. that $V_{\mathcal{S}\cdot}$ contains the first s rows of V and its complement $V_{\bar{\mathcal{S}}\cdot}$ contains the leftover indices, the constraint $V^T M V = I$ in Eq. (4.13) can be reorganized as

$$\begin{bmatrix} V_{\mathcal{S}\cdot} \\ V_{\bar{\mathcal{S}}\cdot} \end{bmatrix}^T \begin{bmatrix} M_{\mathcal{S}\mathcal{S}} & M_{\bar{\mathcal{S}}\mathcal{S}}^T \\ M_{\bar{\mathcal{S}}\mathcal{S}} & M_{\bar{\mathcal{S}}\bar{\mathcal{S}}} \end{bmatrix} \begin{bmatrix} V_{\mathcal{S}\cdot} \\ V_{\bar{\mathcal{S}}\cdot} \end{bmatrix} = I. \tag{4.14}$$

Rearranging it to move all the fixed variables on one side results in

$$V_{\mathcal{S}\cdot}^T M_{\mathcal{S}\mathcal{S}} V_{\mathcal{S}\cdot} + V_{\bar{\mathcal{S}}\cdot}^T M_{\bar{\mathcal{S}}\mathcal{S}} V_{\mathcal{S}\cdot} + V_{\mathcal{S}\cdot}^T M_{\bar{\mathcal{S}}\mathcal{S}}^T V_{\bar{\mathcal{S}}\cdot} =: \hat{H}, \tag{4.15}$$

for a constant matrix $\hat{H}$. With the full-rank assumption on $M_{\mathcal{S}\mathcal{S}}$, completing the square results the following:

$$\left(M_{\mathcal{S}\mathcal{S}}^{\frac{1}{2}} V_{\mathcal{S}\cdot} + M_{\mathcal{S}\mathcal{S}}^{-\frac{1}{2}} M_{\bar{\mathcal{S}}\mathcal{S}}^T V_{\bar{\mathcal{S}}\cdot} \right)^T \left(M_{\mathcal{S}\mathcal{S}}^{\frac{1}{2}} V_{\mathcal{S}\cdot} + M_{\mathcal{S}\mathcal{S}}^{-\frac{1}{2}} M_{\bar{\mathcal{S}}\mathcal{S}}^T V_{\bar{\mathcal{S}}\cdot} \right) = H \tag{4.16}$$

for a new constant matrix $H = \hat{H} + V_{\bar{\mathcal{S}}\cdot}^T M_{\bar{\mathcal{S}}\mathcal{S}} M_{\mathcal{S}\mathcal{S}}^{-1} M_{\mathcal{S}\bar{\mathcal{S}}} V_{\bar{\mathcal{S}}\cdot}$. Since we assume that M is positive definite, $M_{\mathcal{S}\mathcal{S}}$ is also positive definite and invertible.

Now, given an orthogonal subproblem decision matrix P, the next feasible iterate can be provided as

$$V_{\mathcal{S}\cdot} = M_{\mathcal{S}\mathcal{S}}^{-\frac{1}{2}} P H^{\frac{1}{2}} - M_{\mathcal{S}\mathcal{S}}^{-1} M_{\bar{\mathcal{S}}\mathcal{S}}^T V_{\bar{\mathcal{S}}\cdot}, \tag{4.17}$$

which satisfies the constraints in (4.14). Note that by using a *retraction*, we can smoothly map the tangent vectors to the manifold and preserve the key properties of the exponential function necessary to perform feasible descent on the manifold. For the Stiefel manifold, a computationally efficient retraction comes from the Cayley transform which can be extended to the generalized Stiefel manifold shown by Equation (1.2) and Lemma 4.1 of (Wen and Yin, 2013). Consequently, we can eliminate the extra computation of the matrix square roots and simplify (4.17) to the following:

$$V_{S\cdot} = P - M_{SS}^{-1} M_{\bar{S}S}^{T} V_{\bar{S}\cdot}. \tag{4.18}$$

If $P^T M_{SS} P = H$ for $M_{SS} \succ 0$ and non-singular H, the above equation satisfies the subproblem constraint. Thus, given the previous V, the descent curve starts at the point:

$$P_0 = V_{S\cdot} + M_{SS}^{-\frac{1}{2}} M_{\bar{S}S}^{T} V_{\bar{S}\cdot}. \tag{4.19}$$

So far, we have shown how to setup the initial point for the line search step. Next, we describe how to compute the descent curve of the subproblem on the generalized Stiefel manifold for the line search on the manifold. The first step is to find the gradient of the objective function of Eq. (4.13) which is $f(V) = V^T X V + \lambda\, g(V)$. Thus, for $f(U)$ and $V(P)$ where the next feasible point V as a function of P as in Eq. (4.18), the gradient of $f \circ V(P)$ w.r.t. P is

$$\frac{\partial\,(f \circ V(P))}{\partial P} = 2(X_{SS} V_{S\cdot} + X_{S\bar{S}} V_{\bar{S}\cdot}) + \lambda g'(V(P)), \tag{4.20}$$

where $g'(V(P))$ is the subgradient of the regularization term Eq. (4.10). Next, we project the descending subgradient $-G'$ at P_0 onto the tangent space of the manifold $GF_{i,p}(M_{SS})$ by constructing a skew-symmetric matrix:

$$Q = G P_0^T - P_0 G^T, \tag{4.21}$$

which conveniently allows the Cayley transform as in (Wen and Yin, 2013) to smoothly map from the tangent space to the generalized Stiefel manifold

Algorithm 3 Coupled bases framework using SBCD

Given:
$f : GF_{n,p} \to \mathbb{R}$, $V_{[:,:]} \in GF_{n,p}(M_{[:,:]})$, $M_{[:,:]} \in \mathbb{R}^{n \times n}$
while Convergence criteria not met **do**
  for $i = 1, \ldots, K$ **do**
    for $j = 1, \ldots, T$ **do**
      $V_{[i,j]}$:= Free variable
      $V_{[i,j]}$:= SBCD($V_{[i,j]}$) (Alg. 2)
    end for
    for $j = 1, \ldots, T$ **do**
      $R_{[i,j]}$:= Rotation matrix (4.3)
    end for
    for $j = 1, \ldots, T$ **do**
      $M_{[i,j]}$:= Mass matrix (4.6), (4.7)
    end for
  end for
end while

to create the descent curve W as a function of τ:

$$W(\tau) = \left(I + \frac{\tau}{2} Q M_{SS}\right)^{-1} \left(I - \frac{\tau}{2} Q M_{SS}\right) P_0. \tag{4.22}$$

We can linearly search over the descent curve to find the new point $P = W(\tau)$ for some τ which results sufficient decrease in f. Thus, the next feasible decision variable $V' \in \mathbb{R}^{n \times p}$ as a function of P is

$$V'(P) = \begin{bmatrix} P - M_{\bar{S}\bar{S}}^{-1} M_{\bar{S}S}^{T} V_{\bar{S}\cdot} \\ V_{\bar{S}\cdot} \end{bmatrix} \tag{4.23}$$

so we can finally assign the current V with V'. By the construction of the coupled bases model Eq. (4.13), V' is updated to minimize the objective function *while it remains coupled* with the other longitudinal and cross-sectional bases connected to V. The pseudo-code of the algorithm is in Alg. 2.

Iterative SBCD in $GF_{n,p}$ for Bases Coupling.

With the stochastic block gradient descent (SBCD) method roughly similar to (Liu et al., 2015) as a solver for a single coupled basis, we now setup the framework to solve for all coupled bases. Specifically, given multiple matrices $X_{[i,j]}$ for $i = 1, \ldots, K$ and $j = 1, \ldots, T$, we set up grid-like iterations as shown in Alg. 3 where we iteratively pass through all possible pairs of i and j. In each iteration, we set $V_{[i,j]}$ for the current pair of i and j to be the free variable and set $V_{[i',j']}$ of the remaining $i' \neq i$ and $j' \neq j$ to be the fixed variables. We solve for only the free variable $V_{[i,j]}$ using SBCD iteratively for all i and j.

Since Eq. (4.13) imposes the longitudinal coupling based on the mass matrices that are precomputed from the bases available at that iteration, they might not reflect the most accurate longitudinal relations precomputed mass matrices involving rotation matrices of the bases. Therefore, we must update the mass matrices so that they reflect the most accurate bases (encoding the naturally derived longitudinal trajectory) by recomputing the rotation matrices of the newly updated bases. Thus, the mass matrix computation step immediately follows the bases update step. We repeat these steps iteratively until convergence criteria are met.

Scaling of Stochastic Block Coordinate Descent

Below, Table 4.1 shows the average runtimes of the Stochastic Block Coordinate Descent (SBCD) without a regularizer for various $n \times n$ random matrices which essentially solves for the solutions of the generalized eigenvalue problems. This also gives a rough estimate of the runtime of the entire framework which involves computing multiple SBCD operations for all the partitions.

n	100	500	1000	3000	5000	7000	10000	12000
runtime (sec)	0.32	0.44	2.31	34.95	125.78	428.71	979.92	1372.83

Table 4.1: Average runtime of 10 SBCD operations (without a regularizer) for solving $V \in \mathbb{R}^{n\times p}$ given a $n \times n$ matrix X for $p = 20$. The iteration terminated when the objective value is with $< 5\%$ of the true objective value of GEVP.

Singularity Correction

Thus far, we have been assuming that H in Eq. (4.16) derived from the newly represented constraint is nonsingular, allowing us to perform inversions in the subsequent steps. However, even if the initial H is nonsingular, we cannot guarantee to maintain its nonsingularity throughout the iterations. Thus, we now relax that assumption in the previous procedures to consider singular H in the subproblems.

First, we factor out the submatrix of our interest, $M_{\mathcal{S}\mathcal{S}}$ of size $s \times p$, from (16) and rewrite it as

$$\left(V_{\mathcal{S}\cdot} + M_{\mathcal{S}\mathcal{S}}^{-1} M_{\bar{\mathcal{S}}\mathcal{S}}^{T} V_{\bar{\mathcal{S}}\cdot}\right)^{T} M_{\mathcal{S}\mathcal{S}} \left(V_{\mathcal{S}\cdot} + M_{\mathcal{S}\mathcal{S}}^{-1} M_{\bar{\mathcal{S}}\mathcal{S}}^{T} V_{\bar{\mathcal{S}}\cdot}\right) = H. \tag{4.24}$$

Then, iff $V_{\mathcal{S}\cdot} + M_{\mathcal{S}\mathcal{S}}^{-1} M_{\bar{\mathcal{S}}\mathcal{S}}^{T} V_{\bar{\mathcal{S}}\cdot}$ is nonsingular, H will be nonsingular for any submatrix $M_{\mathcal{S}\mathcal{S}}$. Thus, to apply the *singularity correction*, we rearrange the columns of $M_{\mathcal{S}\mathcal{S}}$ to get a new submatrix containing the maximal set of linearly independent columns of the subproblem adjacently. First, assume w.l.o.g. that

$$V_{\mathcal{S}\cdot} + M_{\mathcal{S}\mathcal{S}}^{-1} M_{\bar{\mathcal{S}}\mathcal{S}}^{T} V_{\bar{\mathcal{S}}\cdot} = \begin{bmatrix} U & UC \end{bmatrix} \tag{4.25}$$

for a $s \times r$ nonsingular matrix U and a $r \times (p - r)$ matrix C. Then, for $\mathcal{T}$, the indices of the columns of U, we have

$$U = V_{\mathcal{S}\mathcal{T}} + M_{\mathcal{S}\mathcal{S}}^{-1} M_{\bar{\mathcal{S}}\mathcal{S}}^{T} V_{\bar{\mathcal{S}}\mathcal{T}} \tag{4.26}$$

which allows us to write (4.24) as follows:

$$\begin{bmatrix} U & UC \end{bmatrix}^{T} M_{\mathcal{S}\mathcal{S}} \begin{bmatrix} U & UC \end{bmatrix} = \begin{bmatrix} U^{T} M_{\mathcal{S}\mathcal{S}} U & U^{T} M_{\mathcal{S}\mathcal{S}} UC \\ C^{T} U^{T} M_{\mathcal{S}\mathcal{S}} U & C^{T} U^{T} M_{\mathcal{S}\mathcal{S}} UC \end{bmatrix} = H. \tag{4.27}$$

Treating C as a fixed constant, the equality (4.24) can be reduced to consider only the linearly independent columns $\mathcal{T}$ of the submatrix U as

$$\left(V_{\mathcal{S}\mathcal{T}} + M_{\mathcal{S}\mathcal{S}}^{-1} M_{\bar{\mathcal{S}}\mathcal{S}}^{T} V_{\bar{\mathcal{S}}\mathcal{T}}\right)^{T} M_{\mathcal{S}\mathcal{S}} \left(V_{\mathcal{S}\mathcal{T}} + M_{\mathcal{S}\mathcal{S}}^{-1} M_{\bar{\mathcal{S}}\mathcal{S}}^{T} V_{\bar{\mathcal{S}}\mathcal{T}}\right) = H_{\mathcal{T}\mathcal{T}} \tag{4.28}$$

and (4.27) is true iff the above equation is true. Thus, given $U \in \mathbb{R}^{s \times r}$ such that $U^{T} M_{\mathcal{S}\mathcal{S}} U = H_{\mathcal{T}\mathcal{T}}$, the new feasible iterates for linearly independent columns $\mathcal{T}$ and their complement columns $\bar{\mathcal{T}}$ respectively are

$$V_{\mathcal{S}\mathcal{T}} = U - M_{\mathcal{S}\mathcal{S}}^{-1} M_{\bar{\mathcal{S}}\mathcal{S}}^{T} V_{\bar{\mathcal{S}}\mathcal{T}}, \tag{4.29}$$

$$V_{\mathcal{S}\bar{\mathcal{T}}} = UC - M_{\mathcal{S}\mathcal{S}}^{-1} M_{\bar{\mathcal{S}}\mathcal{S}}^{T} V_{\bar{\mathcal{S}}\bar{\mathcal{T}}}. \tag{4.30}$$

We note that the singularity correction is able to preserve the feasibility of the new iterate by first observing that

$$V_{\mathcal{S}\mathcal{T}} + M_{\mathcal{S}\mathcal{S}}^{-1} M_{\bar{\mathcal{S}}\mathcal{S}}^{T} V_{\bar{\mathcal{S}}\cdot} = \begin{bmatrix} U - M_{\mathcal{S}\mathcal{S}}^{-1} M_{\bar{\mathcal{S}}\mathcal{S}}^{T} V_{\bar{\mathcal{S}}\mathcal{T}} & UC - M_{\mathcal{S}\mathcal{S}}^{-1} M_{\bar{\mathcal{S}}\mathcal{S}}^{T} V_{\bar{\mathcal{S}}\bar{\mathcal{T}}} \end{bmatrix} \tag{4.31}$$

$$+ \begin{bmatrix} M_{\mathcal{S}\mathcal{S}}^{-1} M_{\bar{\mathcal{S}}\mathcal{S}}^{T} V_{\bar{\mathcal{S}}\mathcal{T}} & M_{\mathcal{S}\mathcal{S}}^{-1} M_{\bar{\mathcal{S}}\mathcal{S}}^{T} V_{\bar{\mathcal{S}}\bar{\mathcal{T}}} \end{bmatrix} \tag{4.32}$$

$$= \begin{bmatrix} U & UC \end{bmatrix}. \tag{4.33}$$

Thus, for any $M_{\mathcal{S}\mathcal{S}}$, we show it is exactly the same as Eq. (4.27), reaffirming the next iterate feasibility. Note that we still require that $M_{\mathcal{S}\mathcal{S}}$ to be positive definite for any $\mathcal{S}$, but this can be guaranteed by showing that

$$\mathbf{x}^{T} M_{\mathcal{S}\mathcal{S}} \mathbf{x} \overset{\text{w.l.o.g.}}{=} \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix}^{T} \begin{bmatrix} M_{\mathcal{S}\mathcal{S}} & M_{\bar{\mathcal{S}}\mathcal{S}}^{T} \\ M_{\bar{\mathcal{S}}\mathcal{S}} & M_{\bar{\mathcal{S}}\bar{\mathcal{S}}} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix} \geqslant 0.$$

4.4 Experiments

Data and Scientific Goal

Our dataset focuses on a cohort of middle aged individuals who are at risk for Alzheimer's disease (AD) due to a positive family history (at least one parent with confirmed diagnosis of AD). Our data corresponds to at least

three longitudinal scans of these subjects. The participants are *cognitively healthy* but some AD related brain changes, while subtle, have already begun (Hwang et al., 2019a). Note that in the analysis of anatomical changes in standard magnetic resonance images (*not* brain connectivity as we do here), numerous findings suggest that accurate quantification of brain "changes", e.g., via tensor based morphometry, is often more sensitive than the analysis of individual images independently (Hua et al., 2008). But in the context of AD (and for other brain disorders) most, if not all, such studies focus on data that cover the *full spectrum of the disease* (healthy to AD) – the disease effects of those data are *much stronger* and arguably easier to detect than those in the setting we consider here. We expect that estimating the longitudinal change process in connectivity accurately via the coupled harmonic bases model will enable identifying a *disease signal* even in the pool of *healthy* (but at-risk) individuals.

Deriving Brain Connectivity Networks

There are three key steps in deriving brain connectivity networks for a given population study. **(a) Coordinate system.** For population level analysis of brain images, one typically needs to register all the images onto a standard coordinate system in a way that avoids any unwanted biases. We follow recommended procedures for deriving an unbiased coordinate system for the 3D+time regime as follows (Keihaninejad et al., 2013). We first estimate a subject-specific average that is temporally unbiased. The subject specific averages are then used to generate an unbiased population level average template space, the process is summarized in Fig. 4.3. **(b) Edges.** We use tractography for deriving edges of the *in vivo* brain (structural) networks. The key ingredient needed for these algorithms is the orientational information of white matter fibers passing through a voxel, inferred by fitting a tissue model to the acquired MR signal from diffusion weighted imaging. Our data was acquired at a single

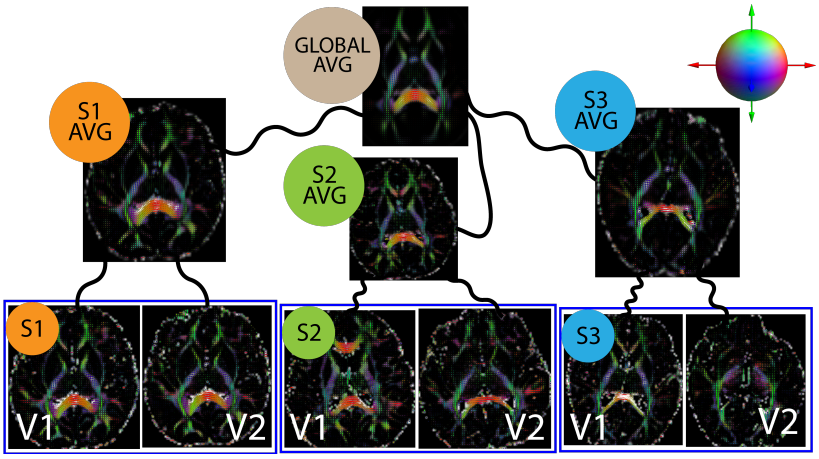


Figure 4.3: Unbiased estimation of the global coordinate system for the longitudinally acquired imaging data. Visits are averaged first which are then used to estimate the global average. Each of the curved black lines represents a combination of rigid, affine and non-linear diffeomorphic transformations. These transformations and the spatial averages are estimated iteratively. Diffusion tensors are directly registered using log-Euclidean framework (Zhang et al., 2007).

shell diffusion weighting of $b = 1000$ s/mm^2. Although limited in its ability to resolve crossing fiber tissue, for this data the most reliable model that can be fit is the so called diffusion tensor. The principal eigenvector of the diffusion tensor in a voxel provides a proxy to the predominant orientation of the white matter fibers in that voxel. Using this information, we repeated probabilistic tractography twenty times (Cook et al., 2006). The normalized standard deviation of the eigenvectors of the tensor (also known as fractional anisotropy) in the tracts passing between two regions serves as the edge strength between nodes in our experiments. **(c) Nodes.** The third key component is the node definition. For defining nodes, we relied on expert neuroanatomy groups who have carefully delineated the boundaries of regions in the brain based on their knowledge (including histological studies). We used the gray matter atlas defined on a DTI template (Varentsova et al., 2014) which provides 160 distinct regions in the brain.

Covariates for Creating Sub-cohorts/partitions

We partition the full cohort using two key ordinal covariates: Rey Auditory Verbal Learning Test (RAVLT) (Schmidt et al., 1996) and Mini Mental State Exam (MMSE) (Tombaugh and McIntyre, 1992). These are cognitive performance scores based on tests that assess the cognitive functioning of the subjects and common in preclinical assessments of AD. Note that there is a systematic effect of age and gender on these scores. To control for these nuisance variables, we perform regression against these variables and derive z-scores for both RAVLT and MMSE. This imposes an implicit ordering of the subjects for the two measures, after the effect of age and sex has been accounted for. We derive K partitions of the z-scores to "stage" the cognitive status (and the subjects) into distinct cognitive quantiles. This implies that even within the full set of "healthy" individuals, subjects that fall within the same quantile are similar. If this staging is finer, we have fewer samples in each partition. We used $\mathsf{K} \in \{2, 3, 4\}$ partitions to keep the individual-level variance manageable while still allowing us to identify the general connectivity evolution patterns. Although we assume that all participants are recruited into the study concurrently (which may not be true), since they are assigned to distinct cognitive quantiles, estimating their longitudinal trajectories is reasonable (also, since the entire cohort is middle aged). In case of uneven distribution along the time axis, standard imputation strategies may be needed. Here, we only utilized data where all three time points were available to keep the presentation simple and avoid concerns related to the potential effect/bias of the specific imputation methods. Once the subjects are assigned to the appropriate partitions based on their z-scores (columns in Fig. 4.2), we can derive the average Laplacians, formulate the system as (4.12) and solve for the coupled bases.

Experiment Design

We use 68 subjects with three longitudinal time points and partition them for different K based on RAVLT and MMSE z-scores. Then, for each setting, we compute four sets of bases: **(a)** non-coupled (4.2), **(b)** longitudinally coupled (4.9), **(c)** cross-sectionally coupled ((4.12) with matrix $M = I$) and **(d)** longitudinally+cross-sectionally coupled (4.12). Thus, each partition now has a set of longitudinal bases (vertical direction in Fig. 4.2). For a novel test subject, we can calculate the corresponding connectivity graph and compare to each of the K partitions. The quantile of the closest partition is the label of this new subject. First, to measure the overall accuracy of this procedure, we use 21 'held out' test subjects, which were *not* used to compute the four sets of bases to avoid overfitting, where each subject has three longitudinal scans available. So, we have total of $21 \times 3 = 63$ distinct Laplacians. We perform two classification tasks. First, we only predict the quantiles of the Laplacians at the *first* (or baseline) time point. Next, we predict all 63 Laplacians which is expected to be much harder since the quantiles of the subsequent time points are not used in deriving the partitions. Nonetheless, we expect that if our coupled bases are accurate, the information from the first time point should, in principle, affect subsequent time points in a way that allows our model to still predict the label correctly. We evaluated $p \in \{n/4, n/2, 3n/4\}$ and chose $p = n/4$ since the residual is extremely small beyond $n/2$. We used $|\mathcal{S}| = 40$, but for larger datasets, it can be set to be larger if computationally feasible while also considering the approximation/speed trade-off. Lastly, we used $\lambda \in \{1, 20, 50\}$.

Results

We show the prediction accuracies of our algorithm in Table 4.2. We run the classification tasks for $K \in \{2, 3, 4\}$ respectively. We compare the

K	Non-coupled		Longitudinal		Cross-section		Coupled	
	j=1	$\{1,2,3\}$	1	$\{1,2,3\}$	1	$\{1,2,3\}$	1	$\{1,2,3\}$
R:2	33.33	34.92	42.86	42.86	66.67	60.32	**71.43**	**71.43**
R:3	38.10	33.33	52.38	36.51	57.14	44.44	**57.14**	**55.56**
R:4	28.57	28.57	23.81	30.16	30.16	23.81	**47.62**	**34.92**
M:2	42.86	41.27	28.57	30.16	57.14	39.68	**76.19**	**71.43**
M:3	42.86	38.10	47.62	49.21	47.62	46.03	**47.62**	**50.79**
M:4	34.92	28.57	23.81	14.29	19.05	12.70	**47.62**	**28.57**

Table 4.2: Prediction accuracy (%) of RAVLT (R:K $\in \{2,3,4\}$ quantiles) and MMSE (M:K $\in \{2,3,4\}$ quantiles) on j $= 1$ time point and j $= \{1,2,3\}$. Best results are in red.

similarities of the bases of each test subject (using ℓ_2 norm) to the set of bases in each partition to locate the closest one, which is the assigned quantile label of the test subject. In Table 4.2, we show the accuracy results for RAVLT and MMSE using four setups (columns 2 through 5): (**a**) non-coupled, (**b**) longitudinally coupled, (**c**) cross-sectionally coupled and (**d**) both longitudinally and cross-sectionally coupled. For RAVLT, first, we discuss the simplest setting for K $= 2$ (R:2 in Table 4.2). Here, the performance estimates suggest that the first three setups are unable to identify the signal whereas our proposed coupled setup offers accuracy estimates approaching 70%. This trend of the coupled setup improving the accuracy of the non-coupled or partially coupled setup continues for K $= 3$ (R:3) and K $= 4$ (R:4). We observe a very similar trend for the MMSE quantile prediction task suggesting that capturing the full set of longitudinal data in terms of its coupled bases offers significant advantages for predicting subject-level cognitive status.

Discussion

We briefly elaborate on the relevance of these findings. Recall that our dataset is preclinical, i.e., all subjects are healthy. This means that the brain connectivity changes that we seek to capture using our proposed formulation are *extremely subtle*. To appreciate the small effect sizes in

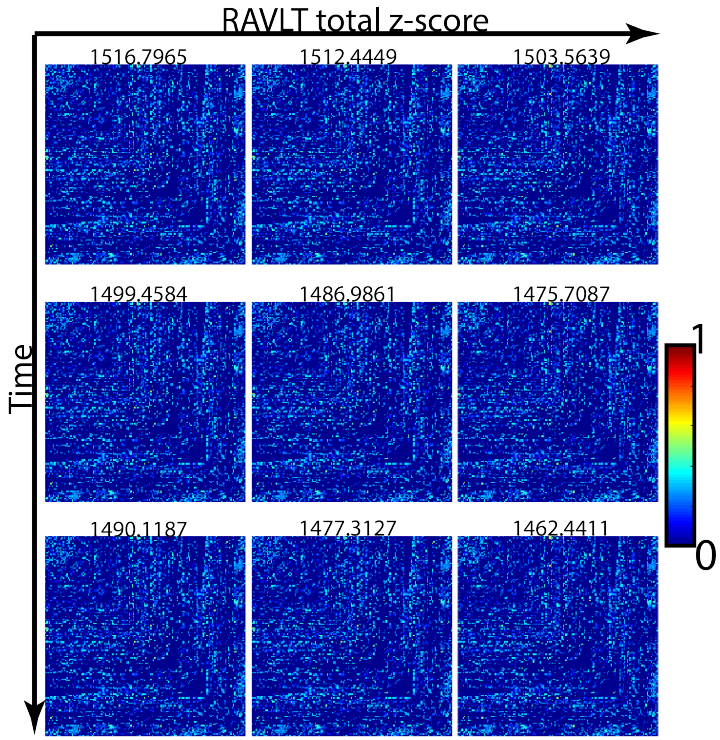


Figure 4.4: Average adjacency matrices for each of the three stages (columns) based on RAVLT total z-scores and each of the three time points (rows). The total magnitude of connectivity strengths (sum of the total edge weights) are shown in the respective titles of the matrices. Even though there is a trend of decrease in the overall connectivity strength along the cognitive staging, the effect sizes are extremely small $0.5 - 1.0\%$. In the case of individual edges the effects are even smaller.

this dataset, we show in Fig. 4.4 the actual brain connectivity adjacency matrices for $K = 3$ from the RAVLT based staging. The numbers at the top of each matrix image is the *total* sum of edge weights in the graph. As expected, when we move from left to right (and top to bottom), the overall connectivity progressively becomes weaker but the changes are *extremely small* and nearly impossible to pick up in a statistically significant way if this analysis were conducted on an edge-by-edge manner. Despite the fact that the overall changes are in the 0.5-1.0% range over the entire set of 12720 edges, our coupled harmonic bases setup is still able to offer better than chance prediction accuracy. Achieving this capability in a preclinical

population is the main scientific result of our experimental evaluations. In the next chapter, we show how the subtle progression of a preclinical cohort can be detected in a unique normative modeling setup.

Finally, we note that our model recovers disease effects at the level of individual tracts reliably: it specifically identified the top 50 fiber tracts with the most changes in connectivity strength across RAVLT progression.

4.5 Summary

The goal of this chapter is to characterize the evolutionary patterns of brain connectivity networks derived from a longitudinal set of middle-aged healthy individuals who are at risk for Alzheimer's disease. The changes in brain connectivity are extremely small during the preclinical stages and existing approaches do not seem to be sensitive enough. We presented a framework which treats the entire set of graph Laplacians of the brain connectivity networks as a system by explicitly considering the coupling between different cognitive as well as the longitudinal (temporal) stages. Our experimental results provide evidence that such a coupled bases approach can indeed provide better insights into the brain network changes across the clinical stages. While the technical development of our framework was motivated by the neuroimaging application, the resultant numerical optimization schemes can be widely applicable for incorporating relevant couplings into generalized eigenvalue problems which are pervasive in many other areas of computer vision and machine learning.

5 SAMPLING-FREE UNCERTAINTY ESTIMATION IN GATED RECURRENT UNITS WITH APPLICATIONS TO NORMATIVE MODELING IN NEUROIMAGING

Thus far, we have demonstrated how to construct a latent representation of data at each time point (i.e., harmonic bases V) that accounts for the sequential patterns within the observed time points. However, in order to make sequential predictions of the future time points *beyond* the observed time points, we need an explicit latent representation (e.g., hidden variables) encoding the temporal pattern that a model can actually use. For instance, in Chapter 1, we described a variant of recurrent neural networks (RNNs) which outputs a hidden representation that explicitly gets incorporated in the subsequent predictions. Also, as we begin utilizing deep models such as RNNs and deriving complex latent representations which are often difficult to qualitatively interpret, we may find it useful to have a way to quantitatively tell how 'certain' the model is about its decision (e.g., confidence level). In this chapter, we aim to develop a sequential neural network model which (1) excels at making long sequential predictions with the *temporal latent variables* (2) while directly quantifying the degree of *uncertainty* of the model parameters including the computed temporal latent variables.

5.1 Overview

Recurrent Neural Networks (RNNs) have achieved state-of-the-art performance in various sequence prediction tasks such as machine translation (Wu et al., 2016b; Jozefowicz et al., 2016), speech recognition (Hinton et al., 2015; Amodei et al., 2016), language models (Cho et al., 2014) as well as medical applications (Jagannatha and Yu, 2016; Esteban et al., 2016). For

sequences with long term dependencies, popular variants of RNN such as Long-Short Term Memory (Gers et al., 1999) and Gated Recurrent Unit (Chung et al., 2014) have shown remarkable effectiveness in dealing with the vanishing gradients problem and have been successfully deployed in a number of applications.

Point estimates, confidence and consequences. Despite the impressive predictive power of RNN models, the predictions rely on the "point estimate" of the parameters. The confidence score can often be overestimated due to overfitting (Fortunato et al., 2017) especially on datasets with insufficient sample sizes. More importantly, in practice, without acknowledging the level of uncertainty about the prediction, the model cannot be entirely trusted in scientific applications. Unexpected performance variations with no sensible way of anticipating this possibility may also be a limitation in terms of regulatory compliance. When a decision made by a model could result in dangerous outcomes in real-life tasks such as an autonomous vehicle not detecting a pedestrian, missing a disease prediction due to some artifacts in a medical image, or radiation therapy misplanning (Lambert et al., 2011), knowing how 'certain' the model is about its decision can offer a chance to look for alternative solutions such as alerting the driver to take over or recommending a different disease test to prevent undesirable outcomes made by erroneous decisions.

Uncertainty. When operating with predictions involving data and some model, there are mainly two sources of unpredictability. First, there may be uncertainty that arises from an imperfect dataset or observations — *aleatoric* uncertainty. Second, the lack of certainty resulting from the model itself (i.e., model parameters) is called *epistemic* uncertainty (Der Kiureghian and Ditlevsen, 2009). Aleatoric uncertainty comes from the observations *externally* such as noise and other factors that cannot typically be inferred systematically. Algorithms instead attempt to calculate *the epistemic uncertainty resulting from the model itself*. This is often also referred to

as *model uncertainty* (Kendall and Gal, 2017).

Related work on uncertainty in Neural networks. The importance of estimating the uncertainty aspect of neural networks (NN) has been acknowledged in the literature. Several early ideas investigated a suite of schemes related to Bayesian neural networks (BNN): Monte Carlo (MC) sampling (MacKay, 1992a), variational inference (Hinton and Van Camp, 1993) and Laplace approximation (MacKay, 1992b). More recent works have focused on efficiently approximating posterior distributions to infer predictive uncertainty. For instance, scalable forms of variational inference approaches (Graves, 2011) suggest estimating the evidence lower bound (ELBO) to efficiently approximate the marginal likelihood of the weights. Similarly, several proposals have extended the variational Bayes approach to perform probabilistic back propagation with assumed density filtering (Hernández-Lobato and Adams, 2015), explicitly update the weights of NN in terms of the distribution parameters (i.e., expectation) (Blundell et al., 2015), or apply stochastic gradient Langevin dynamics (Welling and Teh, 2011) at large scales. These methods, however, theoretically rely on the correctness of the prior distribution, which has shown to be crucial for reasonable predictive uncertainties (Rasmussen and Quinonero-Candela, 2005) and the strength or validity of the assumption (i.e., mean field independence) for computational benefits. An interesting and different perspective on BNN uncertainty based on Monte Carlo dropout was proposed by Gal et al. (Gal and Ghahramani, 2016), wherein the authors approximate the predictive uncertainty by using dropout (Srivastava et al., 2014) at prediction time. This approach can be interpreted as an ensemble method where the predictions based on "multiple networks" with different dropout structures (Lakshminarayanan et al., 2016) yield estimates for uncertainty. However, while the estimated *predictive uncertainty* is less dependent on the data by using a fixed dropout rate independent from the data, uncertainty estimation on the network parameters (i.e. weights)

requires a marginalization over the weight posterior distribution which could be costly and highly dependent on the observed data (Kendall and Gal, 2017; Xiao and Wang, 2019). In summary, while the literature is still in a nascent stage, a number of researchers are studying ways in which uncertainty estimates can be derived for deep architectures similar to those from traditional statistical analysis for various applications (Ribeiro et al., 2018; Sedlmeier et al., 2019).

Other gaps in our knowledge. While the above methods focus on predictive uncertainty, most strategies do not explicitly attempt to estimate the uncertainty of all *intermediate* representations of the network such as neurons, weights, biases and so on. Such information is understandably less attractive in traditional applications, where our interest mainly lies in the prediction made by the final output layer. However, RNN-type sequential NNs often utilize not only the last layer of neurons but also directly operate on the intermediate neurons in making a sequence of predictions (Mikolov et al., 2010). Several Bayesian RNNs have been proposed (Lakshminarayanan et al., 2016; Fortunato et al., 2017) but are based on the BNN models described above. Their deployment is not always feasible under practical time constraints for real-life tasks, especially with high dimensional inputs. Also, stochastic RNN models with stochastic and deterministic layers (Fraccaro et al., 2016) and stochastic state models for reinforcement learning (Gregor et al., 2018) have been proposed, but they do not explicitly estimate the uncertainty of intermediate representations. Further, empirically more powerful variants of RNNs such as LSTMs or GRUs have not been explicitly studied in the literature in the context of uncertainty.

Contributions

In this chapter, our goal is is to enable uncertainty estimation on more powerful sequential neural networks, namely gated recurrent units (GRU),

while addressing the issues discussed above in BNNs. To our knowledge, few (if any) other works offer this capability. We propose a probabilistic GRU, where *all* network parameters follow exponential family distributions. We call this framework the SP-GRU, which operates *directly* on these parameters, inspired in part by an interesting result for non-sequential data (Wang et al., 2016a). Our SP-GRU directly offers the following properties: **(i)** The operations within each cell in the GRU proceed only with respect to the natural parameters *deterministically*. Thus, the overall procedure is completely sampling-free. Such a property is especially appealing for sequential datasets which often suffer from small sample sizes since the sampling procedures for the marginalization of posterior is high dependent on the observed data. **(ii)** Because weights and biases and *all intermediate neurons* of SP-GRU can be expressed in terms of a distribution, their uncertainty estimates can be directly inferred from the network itself. **(iii)** We focus on some well-known exponential family distributions (i.e., Gaussian, Gamma) which have nice characteristics that can be appropriately chosen with minimal modifications to the operations depending on the application of interest. **(iv)** We show how SP-GRU can be used on neuroimaging data for detecting early disease progression in an asymptomatic Alzheimer's disease cohort, in a manner different from the last chapter.

5.2 Recurrent Neural Networks and Exponential Families in Networks

Recurrent Neural Networks

The Gated Recurrent Unit (GRU) and the Long-Short Term Memory (LSTM) are popular variants of RNN where the network parameters are shared across layers. While they both deal with the exploding/vanishing

gradient issues with *cell* structures of similar forms, the GRU does not represent the cell state and hidden state separately. Specifically, its updates take the following form (order of operation is (1) Reset Gate and Update Gate, (2) State Candidate and (3) Cell State):

$$\begin{aligned}
\textbf{Reset Gate: } & r^t = \sigma(W_r x^t + b_r) \\
\textbf{Update Gate: } & z^t = \sigma(W_z x^t + b_z) \\
\textbf{State Candidate: } & \hat{h}^t = \tanh(U_{\hat{h}} x^t + W_{\hat{h}}(r^t \odot h^{t-1}) + b_{\hat{h}}) \\
\textbf{Cell State: } & h^t = (1 - z^t) \odot \hat{h}^t + z^t \odot h^{t-1}
\end{aligned}$$

where $W_{\{r,z,\hat{h}\}}$ and $b_{\{r,z,\hat{h}\}}$ are the weights and biases respectively for their corresponding updates, and x^t and h^t are the input variables and hidden states at time point t respectively. Typical implementations of both GRUs and LSTMs include an output layer outside of the cell to produce the desired outputs. However, they do not naturally admit more than point estimates of hidden states and outputs.

Exponential Families in Networks

In statistics, the properties of distributions within *exponential families* have been very well studied.

Definition 5.1. Let $x \in X$ be a random variable with probability density (or mass) function (pdf/pmf) f_X. Then f_X is an *exponential family distribution* if

$$f_X(x|\eta) = h(x) \exp(\eta^T T(x) - A(\eta)) \tag{5.1}$$

with natural parameters η, base measure $h(x)$, and sufficient statistics $T(x)$. A constant $A(\eta)$ (log-partition function) ensures that the distribution normalizes to 1.

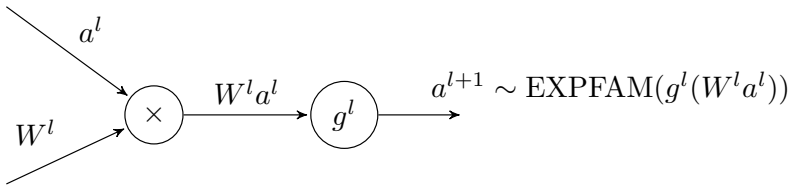


Figure 5.1: A single exponential family neuron. W^l are learned, and the output is a sample generated from the exponential family defined by $g^l(W^l a^l)$.

Common distributions (e.g., Gaussian, Bernoulli, Gamma) can be written in this unified 'natural form' with specific definitions of $h(x)$, $T(x)$ and $A(\eta)$ (e.g., Gaussian distribution with $\eta = (\alpha, \beta)$, $T(x) = (x, x^2)$ and $h(x) = 1/\sqrt{2\pi}$).

Two key properties of this family of distributions have led to their widespread use: (1) their ability to summarize arbitrary amounts of data $x \sim f_X$ through only their sufficient statistics $T(x)$, and (2) their ability to be efficiently estimated either directly through a closed form maximum likelihood estimator or a convex function with convex constraints.

Deep Exponential Families (DEFs) (Ranganath et al., 2015) explicitly model the output of any given layer as a random variable, sampled from an exponential family defined by natural parameters given by the linear product of the previous layer's output and a learnable weight matrix (see Fig. 5.1). While this formulation leads directly to distributions over hidden states and model outputs, we have not learned the distributions over the *model parameters*. We note that even the posteriors that we estimate are "conditioned" on several assumptions. First, we make the mean field assumption on the model parameters which is convenient but does not accurately reflect the true posterior. Second, we often assume the parameters to follow some tractable distributions (e.g., Gaussian in variational inference). Thus, the posterior that we estimate (i.e., $p(W|X, Y)$ for weights W and data X, Y) is actually still within the boundaries of our assumptions, and there is a great deal of effort to minimize the assumptions and

to better approximate the "true" posterior (Myshkov and Julier, 2016; Wu et al., 2019). Computational feasibility is also neglected: the variational inference procedure used for learning these DEFs requires *Monte Carlo sampling at each hidden state many times for every input sample* (the cost of running just *text* experiments was $40K as stated by the authors). This also becomes a concern in many biomedical applications (e.g., medical imaging) where the model size grows proportionally to the dimensionality of data which often ranges from thousands to millions.

5.3 Sampling-free Probabilistic Networks

We now describe a probabilistic network fully operating on a set of natural parameters of exponential family distributions in a *sampling-free* manner. Inspired by a result from a few years back (Wang et al., 2016a), the learning process, similar to traditional NNs, is deterministic yet still captures the probabilistic aspect of the output *and the network itself*, purely as a byproduct of typical NN procedures (i.e., back propagation).

Unlike the probabilistic networks mentioned before, our GRU performs forward propagation in a series of *deterministic* linear and nonlinear transformations on the distribution of weights and biases. Throughout the entire process, all operations only involve distribution parameters while maintaining their desired distributions after every transformation. We focus on three exponential family distributions with two natural parameters: Gaussian, Gamma and Poisson.

Linear Transformations

We describe the linear transformation on the input vector x with a matrix W of weights and a vector b of biases in terms of their natural parameters. We first apply the *mean-field* assumption on each of the weights and biases based on their individual distribution parameters α and β as

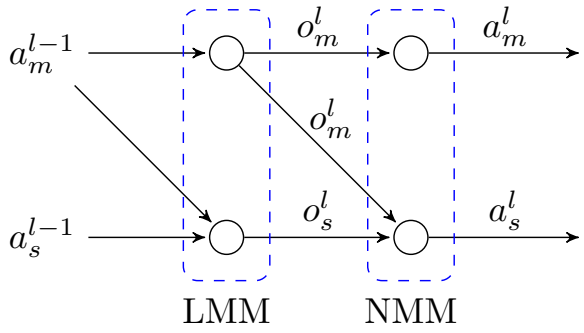


Figure 5.2: Linear Moment Matching (LMM) and Nonlinear Moment Matching (NMM) are performed at the weights/bias sums and activations respectively.

$p(W|W_\alpha, W_\beta) = \prod_{i,j} p(W(i,j) \mid W_\alpha(i,j), W_\beta(i,j))$ where $\{W_\alpha, W_\beta\}$ and $\{b_\alpha, b_\beta\}$ are the model parameters. Thus, analogous to the linear transformation $o = Wa + b$ in ordinary neural networks on the previous layer output (or an input) a with W and b, our network operates purely on (α, β) to compute (o_α, o_β).

After each linear transformation, it is necessary to preserve the 'distribution property' of the outputs (i.e., o_α and o_β still define the same distribution) throughout the forward propagation so that the intermediate nodes and the network itself can be naturally interpreted in terms of their distributions. Thus, we *cannot* simply mimic the typical linear transformation on a_β and compute $o_\beta = W_\beta a_\beta + b_\beta$ if we want o_β to still be able to preserve the distribution (Wang et al., 2016a).

We perform a second order moment matching on the mean and variance of the distributions. The mean m and variance s can easily be computed with an appropriate function $g(\cdot,\cdot)$ which maps $g : (\alpha, \beta) \to (m, s)$ for each exponential family distribution of interest (i.e., $g(\alpha, \beta) = (-\frac{\alpha+1}{\beta}, \frac{\alpha+1}{\beta^2})$ for a Gamma distribution). Thus, we compute the (m, s) counterparts of all the (α, β)-based components (i.e., $(o_m, o_s) = g(o_\alpha, o_\beta)$).

Using the linear output before the activation function, we can now apply Linear Moment Matching (LMM) on (1) the mean a_m following the standard linearity of random variable expectations and (2) the variance

a_s as follows:

$$o_m = W_m a_m + b_m$$
$$o_s = W_s a_s + b_s + (W_m \odot W_m) a_s + W_s (a_m \odot a_m)$$

where $\odot$ is the Hadamard product. Then, we invert back to $(o_\alpha, o_\beta) = g^{-1}(o_m, o_s)$. For the exponential family distributions involving at most two natural parameters, matching the first two moments is sufficient.

Nonlinear Transformations

The next key step in NNs is the element-wise nonlinear transformation where we want to apply a nonlinear function $f(\cdot)$ to the linear transformation output o parametrized by $\eta = (o_\alpha, o_\beta)$. This is equivalent to a general random variable transformation given the probability density function (pdf) p_O for O to derive the pdf p_A of A transformed by $a = f(o)$: $p_A(a) = p_O(f(o))|f'(o)|$. We note that this *change of variable* will reappear in Chapter 6 where $f(\cdot)$ is a unique invertible neural network for a problem of density estimation.

However, well-known nonlinear functions $f(\cdot)$ such as sigmoids and hyperbolic tangents cannot directly be utilized on (o_α, o_β) because the resulting $a = f(o)$ may not be from the same exponential family distribution. Thus, we perform another second order moment matching in terms of mean o_m and variance o_s via Nonlinear Moment Matching (NMM). Ideally, we need to marginalize over a distribution of o given (o_α, o_β) to compute $a_m = \int f(o) p_O(o \mid o_\alpha, o_\beta) do$ and the corresponding variance $a_s = \int f(o)^2 p_O(o \mid o_\alpha, o_\beta) do - a_m^2$ which we map back to (a_α, a_β) with an appropriate bijective mapping function $g(\cdot, \cdot)$. However, when the dimension of o grows, the computational burden of integral calculation becomes incredibly more demanding. The closed form approximations described below can efficiently compute the mean and variance of the activation out-

puts a_m and a_s (Wang et al., 2016a). We show these approximations for sigmoids $\sigma(x)$ and hyperbolic tangents $\tanh(x)$ for a Gaussian distribution, as these will become the critical components used in our probabilistic GRU. Here, we use the fact that $\sigma(x) \approx \Phi(\zeta x)$ where $\Phi(\cdot)$ is a probit function and $\zeta = \sqrt{\pi/8}$ is a constant. Then, we can approximate the sigmoid functions for a_m and a_s as

$$a_m \approx \sigma_m(o_m, o_s) = \sigma\left(\frac{o_m}{(1+\zeta^2 o_s)^{\frac{1}{2}}}\right)$$
$$a_s \approx \sigma_s(o_m, o_s) = \sigma\left(\frac{\nu(o_m+\omega)}{(1+\zeta^2\nu^2 o_s)^{\frac{1}{2}}}\right) - a_m^2$$

where $\nu = 4 - 2\sqrt{2}$ and $\omega = -\log(\sqrt{2}+1)$. The hyperbolic tangent can be derived from $\tanh(x) = 2\sigma(2x) - 1$.

Note that other common exponential family distributions do not have obvious ways to make such straightforward approximations. Thus, we use an 'activation-like' mapping $f(x) = a - b\exp(-\gamma d(x))$ where $d(x)$ is an arbitrary activation of choice with appropriate constants a, b and γ of > 0. Nonlinear transformations of Gamma and Poisson distributions can then be formulated in closed form as well (e.g., $a = b = \gamma = 1$ is a good choice).

Nonlinear Transformations of Some Exponential Family Distributions

We show the closed form solutions of the nonlinear transformations (sigmoid and tanh activation functions) for the Gamma, Poisson, and Gaussian distributions.

Gamma Distribution

The probability density function (pdf) of the Gamma distribution given parameters $\alpha > 0$ and $\beta > 0$ is

$$p_X(x \mid \alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1} \exp(-\beta x)$$

for the support $x \in (0, \infty)$ and $\Gamma(\alpha) = \int_0^{\infty} x^{\alpha-1} \exp(-x) dx$. Then, let $c = a = b$, $c > 0$, and $d(x) = x$ to get the 'activation-like' mapping $f(x) = c(1 - \exp(-\gamma x))$. We first perform a nonlinear transformation with respect to m as follows:

$$\begin{aligned}
a_m &= \int f(o) p_O(o \mid o_\alpha, o_\beta) do \\
&= \int_{o=0}^{\infty} c(1 - \exp(-\gamma o)) \frac{{o_\beta}^{o_\alpha}}{\Gamma(o_\alpha)} o^{o_\alpha - 1} \exp(-o_\beta \odot o) do \\
&= c \int_{o=0}^{\infty} \frac{{o_\beta}^{o_\alpha}}{\Gamma(o_\alpha)} o^{o_\alpha - 1} \exp(-o_\beta \odot o) do \\
&\quad - c \int_{0}^{\infty} \frac{{o_\beta}^{o_\alpha}}{\Gamma(o_\alpha)} o^{o_\alpha - 1} \exp(-(\gamma o + o_\beta) \odot o) do \\
&= c \left[1 - \frac{{o_\beta}^{o_\alpha}}{\Gamma(o_\alpha)} \int_{0}^{\infty} o^{o_\alpha - 1} \exp(-(\gamma o + o_\beta) \odot o) do \right] \\
&= c \left[1 - \frac{{o_\beta}^{o_\alpha}}{\Gamma(o_\alpha)} \odot \Gamma(o_\alpha) \odot (o_\beta + \gamma)^{-o_\alpha} \right] \\
&= c \left[1 - \frac{{o_\beta}^{o_\alpha}}{(o_\beta + \gamma)^{o_\alpha}} \right] \\
&= c \left[1 - \left(\frac{o_\beta}{o_\beta + \gamma} \right)^{o_\alpha} \right]
\end{aligned}$$

and for the variance,

$$\begin{aligned}
a_s &= \int f(o)^2 p_O(o \mid o_\alpha, o_\beta)\,do - a_m^2 \\
&= \left[\int_{o=0}^{\infty} c^2(1 - 2\exp(-\gamma o) + \exp(-2\gamma o))\frac{{o_\beta}^{o_\alpha}}{\Gamma(o_\alpha)} o^{o_\alpha - 1}\exp(-o_\beta \odot o)\,do\right] - a_m^2 \\
&= c^2\left[1 - 2\frac{{o_\beta}^{o_\alpha}}{\Gamma(o_\alpha)} \odot \Gamma(o_\alpha) \odot (o_\beta + \gamma)^{-o_\alpha} + \frac{{o_\beta}^{o_\alpha}}{\Gamma(o_\alpha)} \odot \Gamma(o_\alpha) \odot (o_\beta + 2\gamma)^{-o_\alpha}\right] \\
&\quad - a_m^2 \\
&= c^2\left[1 - 2\frac{{o_\beta}^{o_\alpha}}{(o_\beta + \gamma)^{o_\alpha}} + \frac{{o_\beta}^{o_\alpha}}{(o_\beta + 2\gamma)^{o_\alpha}}\right] - a_m^2 \\
&= c^2\left[\left(\frac{o_\beta}{o_\beta + 2\gamma}\right)^{o_\alpha} - \left(\frac{o_\beta}{o_\beta + \gamma}\right)^{2o_\alpha}\right]
\end{aligned}$$

for $c > 0$ and $\gamma > 0$, where $c = 1$ and $\gamma = 1$ are generally good choices that resemble tanh.

Poisson Distribution

The pdf of the Poisson distribution over the support $x \in \{0, 1, 2, \dots\}$ with a parameter $\lambda > 0$ is

$$p_X(x \mid \lambda) = \frac{\lambda^x \exp(-\lambda)}{x!}. \tag{5.2}$$

Then, let $c = a = b$, $c > 0$ and $d(x) = x$ to get the 'activation-like' mapping $f(x) = c(1 - \exp(-\gamma x))$. The nonlinear transformation on o to obtain a_m is as follows:

$$\begin{aligned}
a_m &= \sum_{x=0}^{\infty} f(x) p_O(o \mid o_\alpha, o_\beta) \\
&= \sum_{x=0}^{\infty} c(1 - \exp(-\gamma x))\frac{o_\alpha^x \exp(-o_\alpha)}{x!} \\
&= c\sum_{x=0}^{\infty} \frac{o_\alpha^x \exp(-o_\alpha)}{x!} - c\sum_{0}^{\infty} \exp(-\gamma x)\frac{o_\alpha^x \exp(-o_\alpha)}{x!} \\
&= c - c(\exp(-o_\alpha))\sum_{x=0}^{\infty} \frac{o_\alpha^x \exp(-\gamma x)}{x!} \\
&= c[1 - \exp(-o_\alpha)\exp(\exp(-\gamma)o_\alpha))]
\end{aligned}$$

and for the variance,

$$\begin{aligned}
a_s &= \sum_{x=0}^{\infty} f(x) p_O(o \mid o_\alpha, o_\beta) - a_m^2 \\
&= \left[\sum_{x=0}^{\infty} c^2 (1 - 2\exp(-\gamma x) + \exp(-2\gamma x)) \frac{o_\alpha^x \exp(-o_\alpha)}{x!} \right] - a_m^2 \\
&= c^2 \sum_{x=0}^{\infty} \frac{o_\alpha^x \exp(-o_\alpha)}{x!} - 2c^2 \exp(-o_\alpha) \sum_{x=0}^{\infty} \frac{o_\alpha^x \exp(-\gamma x)}{x!} \\
&\quad + c^2 \exp(-o_\alpha) \sum_{x=0}^{\infty} \frac{o_\alpha^x \exp(-2\gamma x)}{x!} - a_m^2 \\
&= -c^2 \exp(2(\exp(-\gamma) - 1) o_\alpha) + c^2 \exp((\exp(-2\gamma) - 1) o_\alpha) \\
&= c^2 [\exp((\exp(-2\gamma) - 1) o_\alpha) - \exp(2(\exp(-\gamma) - 1) o_\alpha)]
\end{aligned}$$

for $c > 0$ and $\gamma > 0$.

Gaussian Distribution

Here, we provide details of the nonlinear transformation on the Gaussian distribution. Note that our goal is to compute

$$a = \int_{-\infty}^{\infty} f(x) N(x \mid m, s) dx \tag{5.3}$$

for some nonlinear function $f(x)$, mean m and variance s. First, we consider $f(x) = \sigma(x)$. Then, Eq. 5.3 is the logistic-normal integral:

$$a = \int_{-\infty}^{\infty} \sigma(x) N(x \mid m, s) dx = \int_{-\infty}^{\infty} \frac{1}{1 + e^{-x}} \frac{1}{\sqrt{2\pi s}} \exp\left(-\frac{(x - m)^2}{2s} \right) dx$$

which does not have a closed form solution. Now, we use the fact that a probit function

$$\Phi(x) = \int_{-\infty}^{x} N(z \mid 0, 1) dz$$

can be used to approximate a sigmoid function such that

$$\sigma(x) \approx \Phi(\zeta x)$$

for $\zeta^2 = \pi/8$. Further, we know that

$$\int_{-\infty}^{\infty} \Phi(x)\mathrm{N}(x \mid m, s)\mathrm{d}x = \Phi\left(\frac{m}{\sqrt{1+s^2}}\right)$$

so the nonlinear transformation on o with respect to m is

$$a_m = \int \sigma(o)\mathrm{N}(o \mid o_m, \mathrm{diag}(o_s))\mathrm{d}o \approx \Phi\left(\frac{o_m}{\sqrt{\zeta^{-2}+o_s}}\right) \approx \sigma\left(\frac{o_m}{\sqrt{1+\zeta^2 o_s}}\right)$$

for $\zeta^2 = \pi/8$. Similarly, for the variance s, since

$$\sigma(x)^2 = \Phi(\zeta\nu(x+\omega))$$

for $\nu = 4-2\sqrt{2}$ and $\omega = -\log(\sqrt{2}+1)/2$, we see that

$$\begin{aligned} a_s &= \int \sigma(o)^2\mathrm{N}(o \mid o_m, \mathrm{diag}(o_s))\mathrm{d}o - a_m^2 \\ &= \Phi\left(\frac{\nu(o_m+\omega)}{\sqrt{\zeta^{-2}+\nu^2 o_m}}\right) - a_m^2 \approx \sigma\left(\frac{\nu(o_m+\omega)}{\sqrt{1+\zeta^2\nu^2 o_m}}\right) - a_m^2 \end{aligned}$$

for $\zeta^2 = \pi/8$.

The hyperbolic tangent function can be derived in a similar way since $\tanh(x) = 2\sigma(2x) - 1$. Thus, for $f(x) = \tanh(x)$ over the support $x \in (-\infty, \infty)$,

$$\begin{aligned} a_m &= \int \tanh(o)\mathrm{N}(o \mid o_m, \mathrm{diag}(o_s))\mathrm{d}o \\ &= \int (2\sigma(2o)-1)\mathrm{N}(o \mid o_m, \mathrm{diag}(o_s))\mathrm{d}o \\ &= 2\int \sigma(2o)\mathrm{N}(o \mid o_m, \mathrm{diag}(o_s))\mathrm{d}o - \int_{-\infty}^{\infty} \sigma(2o)\mathrm{N}(o \mid o_m, \mathrm{diag}(o_s))\mathrm{d}o \\ &= 2\int \sigma(2o)\mathrm{N}(o \mid o_m, \mathrm{diag}(o_s))\mathrm{d}o - 1 \\ &\approx 2\int \Phi(2o)\mathrm{N}(o \mid o_m, \mathrm{diag}(o_s))\mathrm{d}o - 1 \\ &= 2\Phi\left(\frac{2\zeta o_m}{\sqrt{1+4\zeta^2 o_s}}\right) - 1 \approx 2\sigma\left(\frac{2o_m}{\sqrt{1+4\zeta^2 o_s}}\right) - 1 \\ &= 2\sigma\left(\frac{o_m}{\sqrt{\frac{1}{4}+\zeta^2 o_s}}\right) - 1 \end{aligned}$$

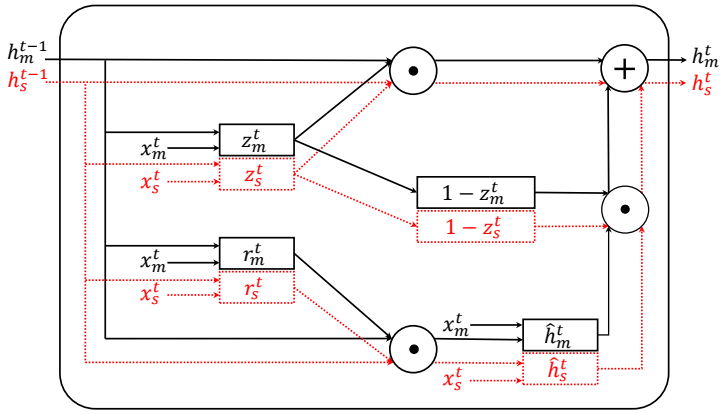


Figure 5.3: SP-GRU cell structure. Solid lines/boxes and red dotted lines/boxes correspond to operations and variables for mean m and variance s respectively. Circles are element-wise operators.

and for the variance,

$$
\begin{aligned}
a_s &= \int \tanh(o)^2 \mathcal{N}(o \mid o_m, \mathrm{diag}(o_s))do - a_m^2 \\
&= \int (4\sigma(2o)^2 - 4\sigma(2o) + 1)\mathcal{N}(o \mid o_m, \mathrm{diag}(o_s))do - a_m^2 \\
&\approx \int (4\Phi(\zeta\nu(o+\omega)) - 4\sigma(2o) + 1)\mathcal{N}(o \mid o_m, \mathrm{diag}(o_s))do - a_m^2 \\
&= \int 4\Phi(\zeta\nu(o+\omega))\mathcal{N}(o \mid o_m, \mathrm{diag}(o_s))do - \int 4\sigma(2o)\mathcal{N}(o \mid o_m, \mathrm{diag}(o_s))do \\
&\quad + 1 - a_m^2 \\
&= 4\Phi\left(\frac{\nu(o_m+\omega)}{\sqrt{\zeta^{-2} + \nu^2 o_s}}\right) - 2\sigma\left(\frac{o_m}{\sqrt{\frac{1}{4} + \zeta^2 o_s}}\right) - 3 - a_m^2 \\
&\approx 4\sigma\left(\frac{\nu(o_m+\omega)}{\sqrt{1 + \zeta^2\nu^2 o_s}}\right) - a_m^2 - 2a_m - 1
\end{aligned}
$$

where $\nu = 2(4 - 2\sqrt{2})$ and $\omega = -\log(\sqrt{2}+1)/2$.

Sampling-free Probabilistic GRU

Based on the probabilistic formulations described above, we present our *Sampling-free Probabilistic GRU* (SP-GRU). The internal architecture is shown

in Fig. 5.3. Here, we focus on adapting GRU with the sampling-free probabilistic formulation. We express all the variables related to the GRU in Table 5.1 in terms of their parameters $\eta = (\alpha, \beta)$. For instance, W_r is now expressed *only* in terms of its parameters $W_{r,\alpha}$ and $W_{r,\beta}$ (i.e., two weight matrices). We assume that all of the variables are factorized. Because the GRU consists of a series of operations with linear and nonlinear transformations, we can update each gate by the transformations defined in Table 5.1.

Assuming that the desired exponential family distribution provides an invertible parameter mapping function $g(\cdot, \cdot)$, we first transform all of the natural parameter variables to means and variances. Then, given an input sequence $x = \{x_m^1, x_s^1\}, \ldots, \{x_m^T, x_s^T\}$, we perform linear/nonlinear transformations with respect to means and variances for each GRU operation (Fig. 5.3 and Table 5.1).

The cell state computation does not involve a nonlinear transformation. For an output layer on the hidden states to compute the desired estimate $\hat{y}$, a typical layer can be defined in a similar manner to obtain both $\hat{y}_m$ and $\hat{y}_s$. In the experiments that follow, we add another such layer to compute the mean and variance of the sequence of predictions $\hat{y} = \{y_m^1, y_s^1\}, \{y_m^2, y_s^2\}, \ldots, \{y_m^T, y_s^T\}$.

Extensibility remarks. We note that despite the simplicity of the cell structure of the SP-GRU as shown in Fig. 5.3, our exponential family adaptation is *not* limited to GRU. For instance, the above formulation can be extended to other variants of RNNs such as LSTMs popularly used in medical applications (Jagannatha and Yu, 2016; Santeramo et al., 2018), flow-based models (Dinh et al., 2016, 2014), and invertible neural networks (Ardizzone et al., 2018). We note that this also extends to our flow-based model with a sequential invertible neural network which we describe in the next chapter.

Operation	Linear Transformation	Nonlinear Transformation
Reset Gate	$o^t_{r,m} = U_{r,m}x^t_m + W_{r,m}h^{t-1}_m + b_{r,m}$ $o^t_{r,s} = U_{r,s}x^t_s + W_{r,s}h^{t-1}_s + b_{r,s} + [U_{r,m}]^2x^t_s$ $+U_{r,s}[x^t_m]^2 + [W_{r,m}]^2h^{t-1}_s + W_{r,s}[h^{t-1}_m]^2$	$r^t_m = \sigma_m(o^t_{r,m}, o^t_{r,s})$ $r^t_s = \sigma_s(o^t_{r,m}, o^t_{r,s})$
Update Gate	$o^t_{z,m} = U_{z,m}x^t_m + W_{z,m}h^{t-1}_m + b_{z,m}$ $o^t_{z,s} = U_{z,s}x^t_s + W_{z,s}h^{t-1}_s + b_{z,s} + [U_{z,m}]^2x^t_s$ $+U_{z,s}[x^t_m]^2 + [W_{z,m}]^2h^{t-1}_s + W_{z,s}[h^{t-1}_m]^2$	$z^t_m = \sigma_m(o^t_{z,m}, o^t_{z,s})$ $z^t_s = \sigma_s(o^t_{z,m}, o^t_{z,s})$
State Candidate	$o^t_{\hat{h},m} = U_{\hat{h},m}x^t_m + W_{\hat{h},m}h^{t-1}_m + b_{\hat{h},m}$ $o^t_{\hat{h},s} = U_{\hat{h},s}x^t_s + [U_{\hat{h},m}]^2x^t_s + U_{\hat{h},s}[x^t_m]^2 + b_{\hat{h},s}$ $+([W_{\hat{h},m}]^2 + W_{\hat{h},s})([r^t_m]^2 \odot h^{t-1}_s$ $+[h^{t-1}_m]^2 \odot r^t_s + r^t_s \odot h^{t-1}_s)$ $+W_{\hat{h},s}([r^t_m]^2 \odot [h^{t-1}_m]^2)$	$\hat{h}^t_m = \tanh_m(o^t_{\hat{h},m}, o^t_{\hat{h},s})$ $\hat{h}^t_s = \tanh_s(o^t_{\hat{h},m}, o^t_{\hat{h},s})$
Cell State	$h^t_m = (1 - z^t_m) \odot \hat{h}^t_m + z^t_m \odot h^{t-1}_m$ $h^t_s = [1 - z^t_m]^2 \odot \hat{h}^t_s + [z^t_m]^2 \odot h^{t-1}_s$ $+z^t_s \odot [\hat{h}^t_m - h^{t-1}_m]^2 + z^t_s \odot [\hat{h}^t_s + h^{t-1}_s]$	Not Needed

Table 5.1: SP-GRU operations in mean and variance. $\odot$ and $[A]^2$ denotes the Hadamard product and $A \odot A$ of a matrix/vector A respectively. Note the Cell State does not involve nonlinear operations. See Fig. 5.3 for the illustration of cell structure.

5.4 Experiments

We first perform unsupervised learning of predicting image sequences from the Moving MNIST dataset (Srivastava et al., 2015) for intuitive quantitative/qualitative evaluations. Second, we apply our model to a unique neuroimaging dataset, consisting of brain imaging acquisitions from individuals at risk for developing Alzheimer's disease. Models were trained on an NVIDIA GeForce GTX 1080 Ti GPU in TensorFlow with ADAM and an initial learning rate of 0.05, and decay parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$. We use the Gaussian distribution for all setups with the KL divergence between the final output distribution $N(o_m, diag(o_s))$ and the target mini-batch distribution $N(y_m, diag(y_s))$ as the error where y_m and y_s are the ground truth values of the mini-batch samples and their variances (w.r.t. the current mini-batch) respectively. This allows the

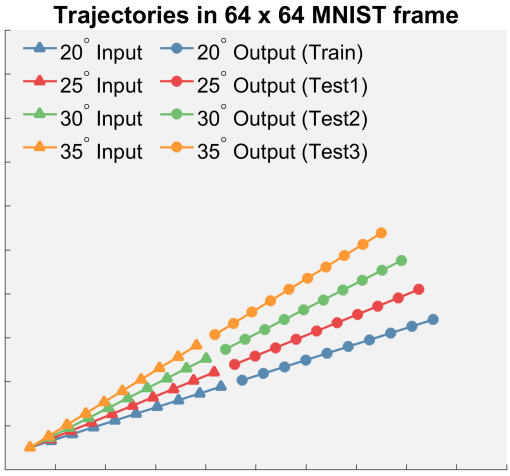


Figure 5.4: Trajectories with training angle (20°), and three test angles (25°, 30° and 35°).

model to learn both the means and variances.

Unsupervised Sequence Learning of Moving MNIST

Controlled Moving MNIST

For pixel-level tasks, prediction quality can be understood by the uncertainty estimate, i.e., estimated model variance of that pixel. In these experiments, we ask the following questions qualitatively and quantitatively: (1) Given a visually 'good looking' sequence prediction, how can we tell that its trajectory is correct? (2) If it is, can we derive a degree of uncertainty on its prediction?

Setup. The moving MNIST dataset consists of digits moving (randomly or controlled) in a 64×64 image over 20 frames. We split sequences into two halves (first 10 and second 10 frames). Then, we encode the first 10 frames to learn a *temporal latent representation* (size 1024) and use this to predict the second 10 frames.

Controlled Paths. We first train our SP-GRU and Monte Carlo dropout GRU (MC-GRU) (Gal and Ghahramani, 2016) with the same number of

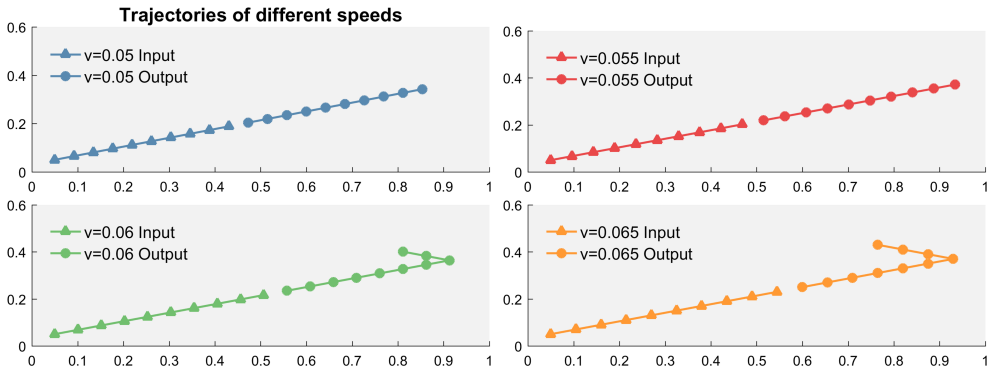


Figure 5.5: Trajectories with (Top left) training speed (5% of image size $= 64 * 0.05 \approx 3.2$ pixels per frame), and (Top right, Bottom left, Bottom right) three test angles (5.5%, 6% and 6.5%).

parameters until they have similar test errors (independent of uncertainty) on simple one-digit MNIST sequences moving in a straight line (blue line in Fig. 5.4). We then construct three sets of 100 'unfamiliar' samples where each set consists of sequences deviating from the training sequence path (blue path in Fig. 5.4 with angle $\theta = 20°$ and speed $v = 5.0\%$ of width per frame) with varying angles (25°, 30°, and 35° paths in Fig. 5.4) and speeds (5.5%, 6.0%, and 6.5% of width per frame in Fig. 5.5).

Results. For 'unfamiliar' angles and speeds, the predictions in Fig. 5.6 look visually sensible, but they do *not* actually follow the ground truth *paths* (e.g., the prediction of 35° still follows 20° path). We can quantify this directly by the [sum of pixel-level variances / frames] as shown in the right of Fig. 5.6. While we cannot evaluate the relative difference here because the 'ground truth uncertainty' is not available for a true comparison, we observe that the *uncertainty increases as the angle/speed deviation increases* for both SP-GRU and MC-GRU.

Computation Speed. From a practical perspective, the uncertainty estimation should not sacrifice computational speed, e.g., real-time safety of an autonomous vehicle. With respect to this crucial aspect, SP-GRU greatly benefits from its *sampling-free* procedure: each epoch (30 sequences) takes ~3 seconds while MC-GRU with a Monte Carlo sampling rate of 50 requires

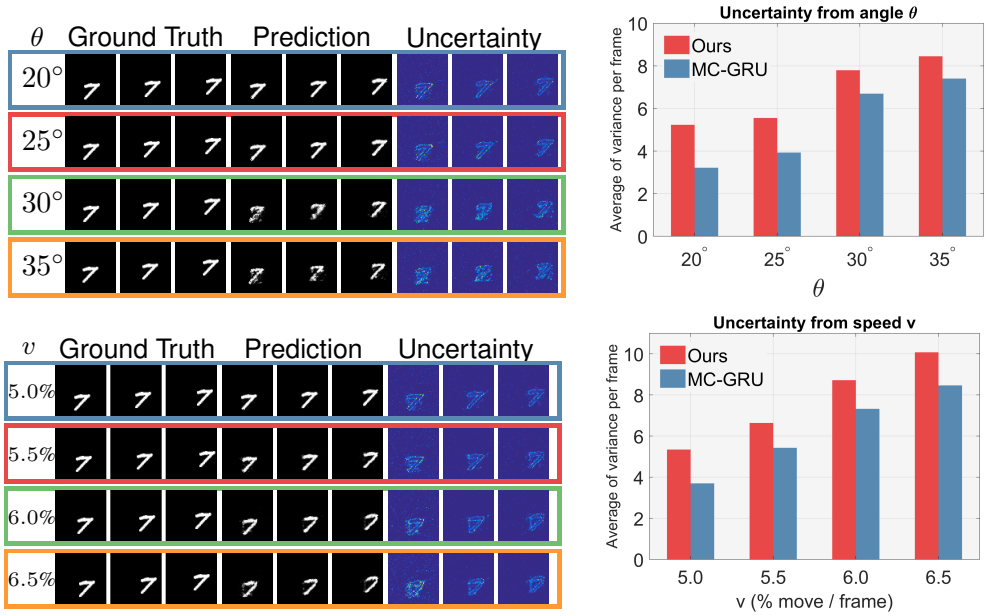


Figure 5.6: Predictions and uncertainties (sub-frames $\{11, 15, 20\}$ out of full predicted frames $\{11, \ldots, 20\}$) from testing varying deviations from trained trajectories (first of four rows, blue). Top: angle (colors match the paths in Fig. 5.4). Bottom: speed (colors match the paths in Fig. 5.5). Right: [sum of pixel-level variances / frames] using SP-GRU and MC-GRU.

~40 seconds ($>$ 10 times SP-GRU) despite their comparable qualitative and quantitative performance. The MC sampling rate for these methods *cannot* simply be decreased which will underestimate the uncertainty. With SP-GRU, we compute this model uncertainty *in a closed form*, without the need for any heavy lifting from large sample analysis. Interestingly, we will see how such measure of confidence could also be estimated as a *density estimation problem of sequential samples* in the next chapter.

Random Moving MNIST

To demonstrate that SP-GRU does not sacrifice the base predictive power (i.e., mean prediction), we evaluate SP-GRU on a public benchmark setup of 2 randomly moving digits (Srivastava et al., 2015).

Results. An example of two digit prediction result is illustrated in

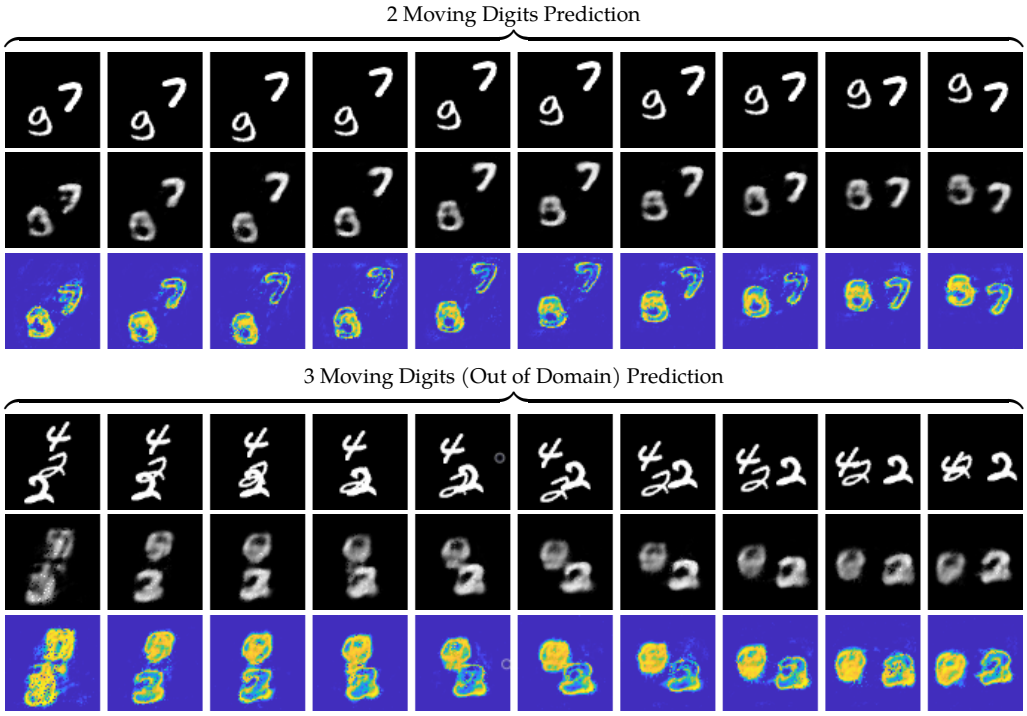


Figure 5.7: SP-GRU predictor results. Top 3 rows: 2 moving digits (top: ground truth, middle: mean prediction, bottom: uncertainty estimate). Bottom 3 rows: 3 moving digits which are out of domain (i.e., not seen in training).

Fig. 5.7 (Top 3 rows) which shows quantifiable variance outputs as demonstrated in the controlled paths examples. We note that the mean prediction (middle row of Top 3 rows in Fig. 5.7) performance is also accurate by comparing our method to previous work in Table 5.2. SP-GRU with a basic predictor network setup performs comparably or better than other methods that do *not* provide model uncertainty. In these works, model performance often benefits from respective specific network structures: encoder-predictor composite models (Srivastava et al., 2015), generative adversarial networks (Ghosh et al., 2016), and external weight filters (De Brabandere et al., 2016). Further, more advanced models (Cricri et al., 2016) have achieved better results with large, more sophisticated pipelines. Extending SP-GRU to such setups becomes a reasonable modification,

Model	Test Loss
Srivastava et al. 2015	341.2
Xingjian et al. 2015	367.1
Brabandere et al. 2016	285.2
Ghosh et al. 2016	241.8
SP-GRU (Ours)	277.1

Table 5.2: Average cross entropy test loss per image per frame on Moving MNIST.

providing model uncertainty *without* sacrificing performance.

We also evaluate how well SP-GRU is able to perform on *out-of-domain* samples (Fig. 5.7, Bottom 3 rows). Models deployed in real-world settings may not realistically be able to determine if a sample is far from their training distributions. However, with our specific modeling of uncertainty, we would expect that images or sequences distant from the training data will exhibit high variance. We construct sequences of 3 moving digits. Here, future reconstruction is generally quite poor. As it has been observed in the previous work (Srivastava et al., 2015), the model attempts to hallucinate only two digits. Our model is *aware of this issue*: the variance for a large number of pixels is extremely high, *even if the digits overlap*. Again, the idea of detecting such "out of domain" samples will reappear in the next chapter which aims to estimate the *density* of the sample with respect to the dataset distribution.

Other Methods

Deep Markov Model (DMM) (Krishnan et al., 2017) is a variant of Structured Variational Autoencoders introduced recently that naturally give rise to a probabilistic interpretation of predictions from deep temporal models. However, upon application of this model to Moving MNIST we were unable to obtain any reasonable prediction, across a range of hidden dimension sizes and trajectory complexities, even with significant training time (days for DMM vs. hours for SP-GRU). Shown in Fig. 5.8 are the

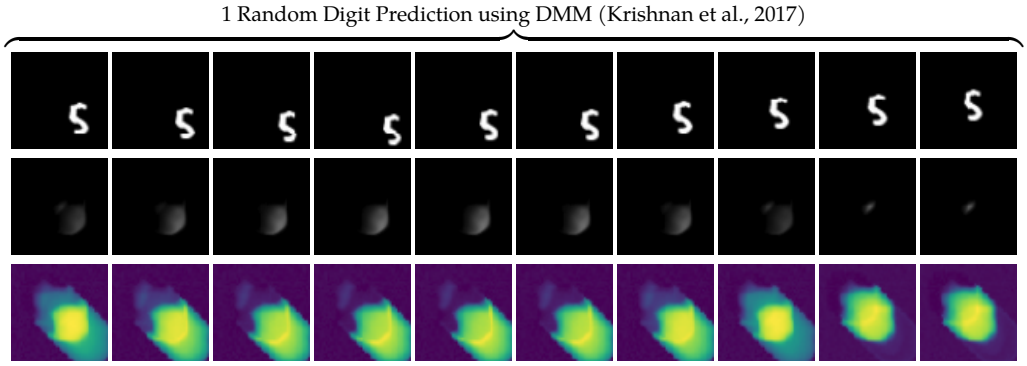


Figure 5.8: Deep Markov Model results. Compared to our results on a single digit (Fig. 5.6), the mean and variance estimations using DMM cannot be estimated well on Moving MNIST.

results using a hidden dimension size of 1024 (equal to our setup). We note that the experimental setups described in (Krishnan et al., 2017) are small in dimension and complexity compared to Moving MNIST, and it may be the case that additional technical development with DMMs may lead to promising and comparable uncertainty results.

Normative Modeling in Preclinical Neuroimaging Data

In a *preclinical cohort* of individuals at risk for developing Alzheimer's disease (AD), effect sizes are small and statistical signal is often weak among those who will and will not go on to develop AD. Even with a high-dimensional brain imaging data, it is often the case that specific imaging modalities do not lead to significant group differences. Early detection of risk factors associated with the eventual development of AD are of critical importance in facilitating the prevention of onset, and identifying individuals who subtly deviate from expected decline is a required step in that direction. We aim to identify an out-of-domain sample via *normative modeling* (Marquand et al., 2016): Given that we have a SP-GRU model trained on a preclinical cohort, can we *predict with confidence* those individuals who are *at risk*?

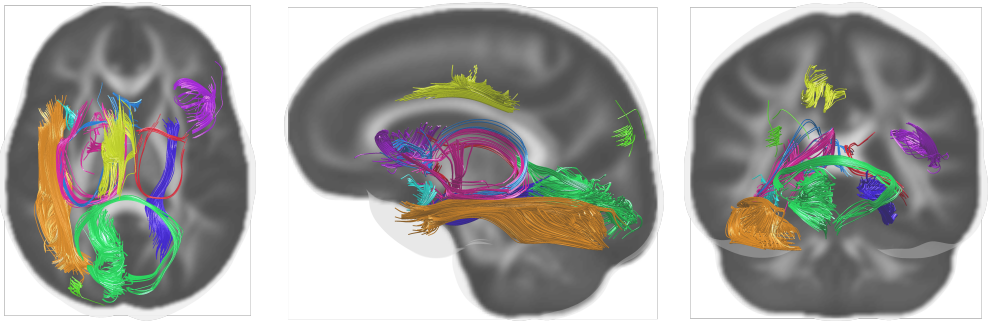


Figure 5.9: Fiber bundles of the brain connectivities known previously to be associated with preclinical cohort developing AD.

Brain Imaging Data of a Preclinical Alzheimer's Disease Cohort

Imaging data from 139 individuals was derived from two distinct modalities: Positron emission tomography (PET) and diffusion-weighted MRI (DWI) (Chua et al., 2008). PET imaging is used to determine mean amyloid-plaque burden (^{11}C Pittsburgh Compound B (PiB) radiotracer), known to be strongly associated with AD pathology and often *preceding* observable cognitive decline (Johnson et al., 2014b). An individual is deemed *at risk* if the average amyloid burden within specific regions (eight bilateral) is greater than 1.12 (Johnson et al., 2014b). For more details about the amyloid pathology and PiB PET scans, see (Johnson et al., 2014b; Hwang et al., 2019a).

DWI captures the diffusion of water through a specific voxel in a brain image; the mean diffusion of water through a *tract* within the brain is a measure of connectivity strength. For each individual, 1761 unique brain connectivities derived from the IIT atlas (Varentsova et al., 2014) are computed from each DWI. The brain connectivity network construction follows Sec. 4.4 in Chapter 4.

Additionally, we have a neuropsychological test score for each individual, the Rey Auditory Visual Learning Test (RAVLT) (Rosenberg et al., 1984) known to be correlated with both amyloid load and structural connectivity. Since our data is cross-sectional, we use RAVLT as our "temporal"

analog of cognitive decline.

Preprocessing Pipeline

To generate our sequential training data, we first place *all* individuals into 8 bins based on their RAVLT scores (i.e., 8 evenly ranged intervals between $[\text{RAVLT}_{\text{max}}, \text{RAVLT}_{\text{min}}]$). This gives us the sample means and variances of each connectivity in each bin. Then, we generate samples of 1761 connectivities across 8 bins (time points) by independently sampling each connectivity in each bin from a normal distribution with the corresponding sample mean and variance. See Fig. 5.10 for the illustration.

To generate the PiB+ (above threshold amyloid burden) and PiB- (below threshold amyloid burden) groups for the test data, we repeat the above data generation process, one with the PiB+ subjects only and the other with the PiB- subjects only. See Fig. 5.11 for the illustration.

Evaluation

We follow existing work in identifying at-risk individuals. Refer to Fig. 5.12 for the full pipeline. First, after we train our SP-GRU predictor, we generate $N = 100$ new test sequences and predict $t = 5, 6, 7, 8$ given $t = 1, 2, 3, 4$ (Fig. 5.12 (1)-(2)). Thus, for subject i, time t and connectivity k, we obtain a mean response $\bar{y}_{itk}$ and an expected level of variation σ_{itk}. Note that we also have the true response y_{itk} with a bin-level variance of σ_{ntk}. Then, we compute a *normative probability map* (NPM) per timepoint for each subject and connectivity (Ziegler et al., 2014). We compute Z-scores across timepoints, connectivities, and subjects as $z_{itk} = (y_{itk} - \bar{y}_{itk})/\sqrt{\sigma_{itk}^2 + \sigma_{ntk}^2}$ (Fig. 5.12 (3)). Applying the procedure described in (Marquand et al., 2016) we compute subject-level empirical distributions of all connectivities per timepoint. Then the robust mean of the top 5% of absolute statistics defines the extreme value statistic (EVS) describing that subject (Fig. 5.12

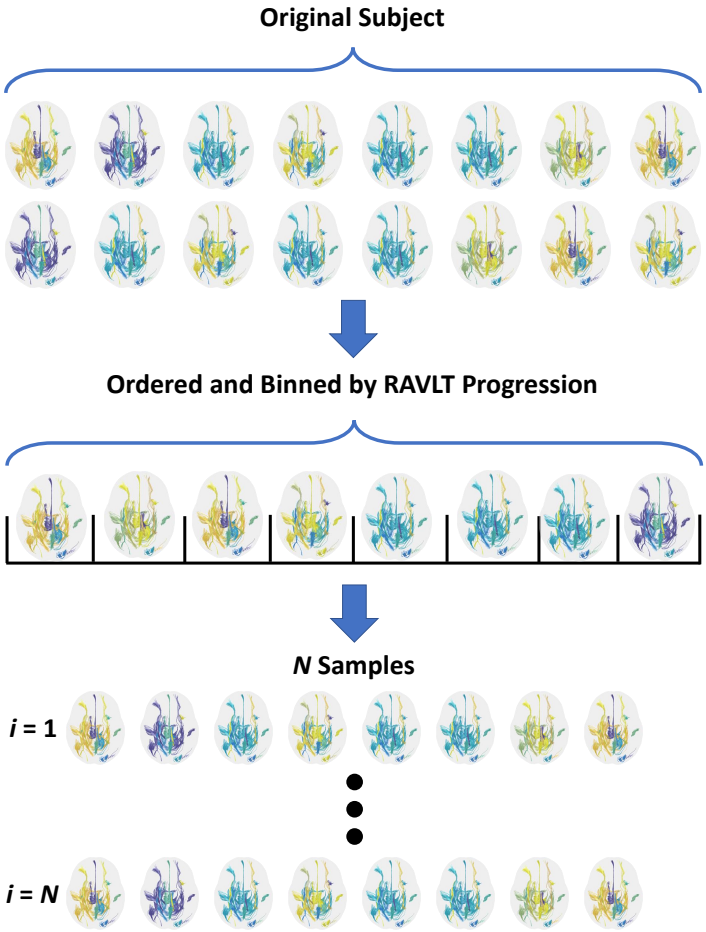


Figure 5.10: Sample data generation of the training set. (1) Place *all* individuals into 8 bins based on their RAVLT scores (i.e., 8 evenly ranged intervals between [$\text{RAVLT}_{\text{max}}$, $\text{RAVLT}_{\text{min}}$]). (2) Compute sample means and variances of each connectivity in each bin. (3) Generate samples of 1761 connectivities across 8 bins (time points) by independently sampling each connectivity in each bin from a normal distribution with the corresponding sample mean and variance.

(4)). Collecting across subjects we fit a generalized extreme value distribution (GED) per time point (Fig. 5.12 (5)).

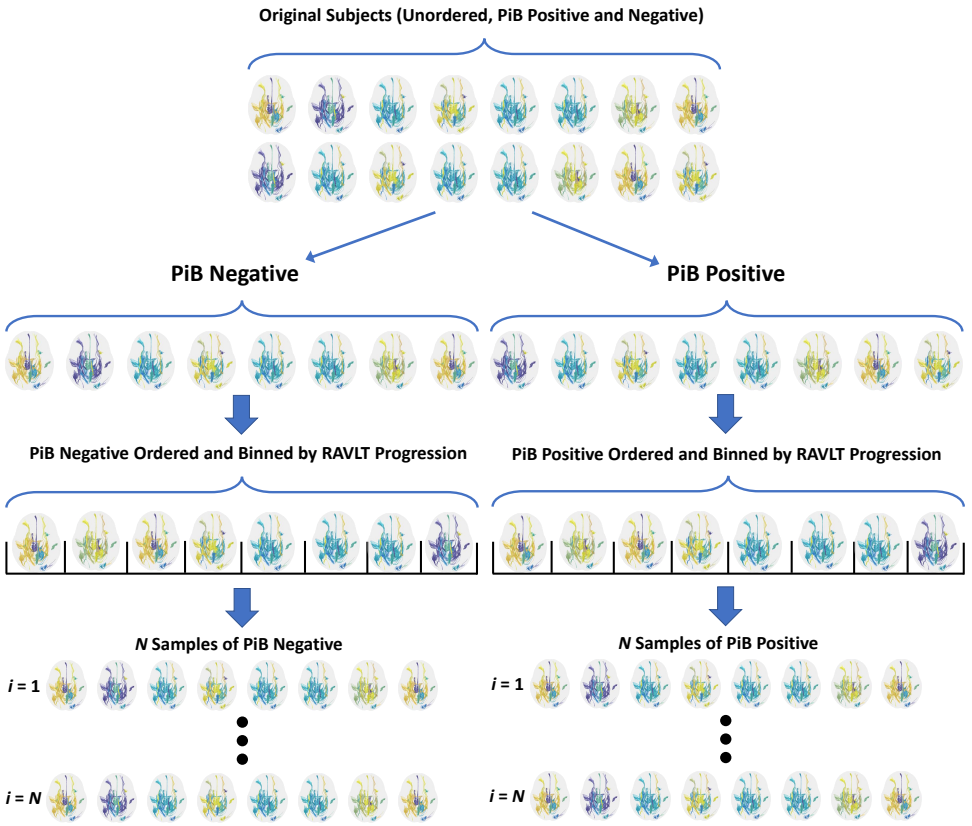


Figure 5.11: Sample data generation of the test set. The procedure is similar to the training set generation described in Fig. 5.10, each with PiB+ and PiB- subject groups.

Results

We aim to identify those sequences which correspond to individuals deviating from the norm defined by our estimated GED. Based on the amyloid burden, we can separate our cohort into two distinct groups, one of which is considered to be 'cognitively healthy' (PiB-), the other to be 'at risk' (PiB+). Sampling 100 sequences each using the binning above applied to both groups, we can then apply the EVS procedure (i.e., compute EVS following (1)-(4) in Fig. 5.12 with the same SP-GRU). Then, we use these EVS to identify sequences within those groups which significantly deviate from the overall population (Fig. 5.12 (6)-(7)). With an $\alpha = 0.01$ cutoff

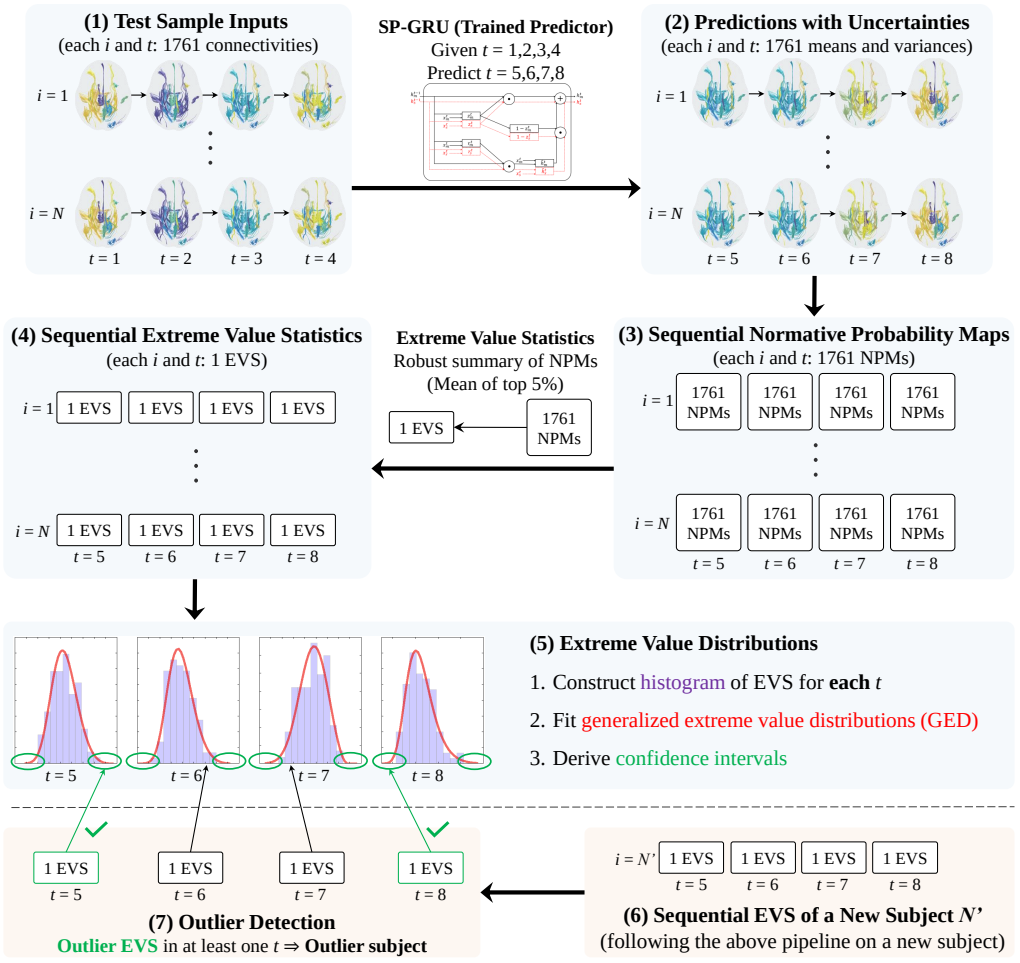


Figure 5.12: Normative modeling pipeline for preclinical AD. **(1)** Given a set of test inputs $(t = 1, 2, 3, 4)$, **(2)** use the pretrained SP-GRU to make mean and variance predictions for each connectivity and $t = 5, 6, 7, 8$. **(3)** Compute NPM for each prediction, and **(4)** derive EVS for each sample i and t. **(5)** Fit GED and construct confidence intervals based on N EVS for each t. **(6)** Given a new sample, derive EVS following (1)-(4), and **(7)** check the confidence intervals from (5) to determine heterogeneity.

(with the Bonferroni correction) we identify 9 outlier sequences in the cognitively healthy group (PiB-) and 19 in the at risk group (PiB+). While further scientific analysis is necessary, these results suggest that larger absolute fluctuations in DWI connectivity may be a good indicator for disease risk as measured by amyloid burden. This sets a promising direc-

tion in preclinical AD research since brain connectivity is one of the early indicators of AD progression (Greicius et al., 2009; Kim et al., 2015, 2019; Hwang et al., 2019a) characterizing the overall integrity of brain.

5.5 Summary

In this work, we show how uncertainty estimates for a powerful class of sequential models, GRUs, can be derived without compromising either predictive power or computation speed using our SP-GRU. Complementary to the developing body of work on Bayesian perspectives on deep learning, we show how a mix of old and new ideas can enable deriving uncertainty estimates for a powerful class of models, GRUs, while also being easily extensible to other sequential models which also derive temporal latent representations. Competitive results are first shown on a standard dataset used for sequential models, while offering uncertainty as a natural byproduct. We then demonstrated a direct application of SP-GRU for normative modeling of preclinical Alzheimer's disease cohort for outlier detection yielding results consistent with the findings in the field. In the upcoming chapters, we will continuously see how other types of temporal latent representation solve diverse problems involving sequential data beyond the prediction task that we demonstrated in this chapter.

6 CONDITIONAL RECURRENT FLOW: CONDITIONAL GENERATION OF LONGITUDINAL SAMPLES WITH APPLICATIONS TO NEUROIMAGING

A number of applications and analyses often search for the relationship between the sequences of multiple modalities. In fact, we witnessed such a case in Chapter 4 when we characterized the brain network progression with respect to a covariate progression via cross-sectional and longitudinal coupling of the harmonic bases. However, as we pointed out in Chapter 5, explicitly deriving the latent representations that encode the relationships via deep models (e.g., h^t from GRU) may enable much more complex tasks (e.g., long sequential predictions). In this chapter, we integrate several types of relationships, namely, the temporal relationship via the latent variables from sequential deep models and, importantly, the relationship between *sequential modalities* which we seek to understand the associations of. Interestingly, using a generative model with an invertible property, we show how we can (1) quantify a degree of confidence of a sample by estimating its *density* with respect to an unknown dataset distribution and (2) generate sequential samples *conditioned* on real world observations or measures.

6.1 Overview

Consider a dataset of *longitudinal or temporal sequences* of data samples $\{\mathbf{x}^t\}_{i=1}^N$ where each sample $\mathbf{x}_i$ comes with *sequential covariates* $\{\mathbf{y}^t\}_{i=1}^N$, one for each time point t. In other words, we assume that for each sequential sample i, $\mathbf{x}_i^1, \cdots, \mathbf{x}_i^T = \{\mathbf{x}^t\}_i$, the sequential covariates $\mathbf{y}_i^1, \cdots, \mathbf{y}_i^T = \{\mathbf{y}^t\}_i$ provide some pertinent auxiliary information associated with that sequential sample. For example, in a neuroimaging study, if the sequential

samples correspond to several longitudinal image scans of a participant over multiple years, the sequential covariate associated with each time point may be an assessment of disease severity or some other clinical measurement. If the sequential data corresponds to heart rate sensors when a participant is watching a video, the sequential covariate may indicate the presence of violence in the corresponding video segment. Our high level goal is to design conditional generative models for such sequential data. In particular, we want a model which provides us a type of flexibility that is highly desirable in this setting. For instance, for a sample drawn from the distribution after the generative model has been estimated, we should be able to "adjust" the sequential covariates, say at a time point t, dynamically to influence the expected future predictions after t for *that* sample. It makes sense that for a heart rate sequence, the appropriate subsequence should be influenced by *when* the "violence" stimulus was introduced *as well as* the default heart rate pattern of the specific sample (participant) (Akselrod et al., 1981). Notice that when $t = 1$, this construction is similar to conditional generative models where the "covariate" or condition $\mathbf{y}$ may simply denote an attribute that we may want to adjust for a sample: for example, increase the smile or age attribute for a face image sampled from the distribution as in (Kingma and Dhariwal, 2018).

We want our formulation to provide a modified set of $\mathbf{x}^t$s adaptively, if we adjust one or more sequential covariates $\mathbf{y}^t$s for that sample. If we know some important clinical information at some point during the study (say, at $t = 5$), this information should influence the future generation $\mathbf{x}^{t>5}$ conditioned both on this sequential covariate or event $\mathbf{y}^5$ as well as the past sequence of this sample $\mathbf{x}^{t<5}$. This will require *conditioning* on the corresponding sequential covariates at *each* time point t by accurately capturing the posterior distribution $p(\mathbf{x}^t|\mathbf{y}^t)$. Such *conditional sequence generation* needs a generative model for a *sequential* data which can dynamically incorporate time-specific *sequential* covariates $\mathbf{y}^t$ of interest to

adaptively modify sequences.

The above setup models a number of applications in medical imaging and computer vision that may need generation of frame sequences conditioned on frame-level covariates. In neuroimaging, many longitudinal studies focus on identifying disease trajectories (Alexander et al., 2002; Baddeley et al., 1991; Landin-Romero et al., 2017): for example, at what point in the future will the brain or specific regions in the brain exceed a threshold for brain atrophy? The future trend is invariably a function of clinical measurements that a participant provides at each visit *as well as* the past trend of the subject. From a methodological standpoint, constructing a sequential generative model may appear feasible by appropriately augmenting the generation process using existing generative models. For example, it seems that one could simply concatenate the sequential measurements $\{\mathbf{x}^t\}$ as a single input for existing non-sequential conditional generative models such as conditional GANs (Mirza and Osindero, 2014; Isola et al., 2017) and conditional variational autoencoders (Sohn et al., 2015; Abbasnejad et al., 2017). We will see why this is not ideal shortly.

We find that for our application, an attractive alternative to discriminator-generator based GANs, is a family of neural networks called normalizing flow (Rippel and Adams, 2013; Rezende and Mohamed, 2015; Dinh et al., 2016, 2014) which involves *invertible networks* (i.e., reconstruct the input from its output). What is particularly relevant is that such formulations work well for *conditionally* generating diverse samples with controllable degrees of freedom (Ardizzone et al., 2018) – with an explicit mechanism to adjust the conditioning variable (or covariate). But the reader will notice that while these models, in principle, can be used to approximate the posterior probability given an input of any dimension, concatenating a series of sequential inputs quickly blows up the size for these highly expressive models and quickly renders them impractical to run, even on high end GPU clusters. Even if we optimistically assume computational

feasibility, variable length sequences cannot easily be adapted to these innately non-sequential generative models, especially for those that extend beyond the training sequence length. Also, the data generated in this manner involves simply "concatenated" sequential data and does not take into account the innate temporal relationships among the sequences which is fundamental in the success of recurrent models.

Given various potential downstream applications and the issues identified above with conditional sequential generation problem, we seek a model which (i) efficiently generates high dimensional sequence samples of *variable lengths* (ii) with dynamic time-specific conditions reflecting upstream observations (iii) with fast posterior density estimation (with respect to the data we observe and the model which derives the density).

Contributions

We tackle the foregoing issues by introducing an invertible recurrent neural network, *CRow*, that includes *recurrent subnetwork* and *temporal context gating*. These modifications are critical in the following sense. *Invertibility* lets us precisely estimate the distribution of $p(\mathbf{x}^t|\mathbf{y}^t)$ in latent space. Introducing *recurrent subnetworks* and *temporal context gating* enables obtaining cues from previous time points $\mathbf{x}^{<t}$ to generate temporally sensible subsequent time points $\mathbf{x}^{\geqslant t}$. Specifically, our contributions are: **(i)** Our model generates conditional sequential samples $\{\mathbf{x}^t\}$ given sequential covariates $\{\mathbf{y}^t\}$ for $t = 1, \ldots, T$ time points where T can be arbitrarily long. Specifically, we allow this by posing the task as a *conditional sequence inverse problem* based on a conditional invertible neural network (Ardizzone et al., 2018). **(ii)** Assessing the quality of the generated samples may not be trivial for certain modalities (e.g., non-visual features). With the specialized capability of the normalizing flow construction, our model estimates the posterior probabilities $p(\mathbf{x}^t|\mathbf{y}^t)$ of the generated sequences at each time point for potential downstream analyses involving uncertainty similar

to what we showed in Chapter 5. **(iii)** We demonstrate an interesting practical application of our model in a longitudinal neuroimaging dataset. We show that the generated longitudinal brain pathology trajectories (an illustration in Fig. 1.8) can lead to identifying specific regions in the brain (as opposed to the brain connectivity measures we have shown in Chapter 4 and Chapter 5) which are statistically associated with Alzheimer's disease.

6.2 Preliminary: Invertible Neural Networks

We first describe an *invertible neural network* (INN) which inverts an output back to its input for solving inverse problems (i.e., $\mathbf{z} = \mathrm{f}(\mathbf{x}) \Leftrightarrow \mathbf{x} = \mathrm{f}^{-1}(\mathbf{z})$). This becomes the building block of our method; thus, before we present our main model, let us briefly describe a specific type of invertible structure which was originally specialized for density estimation with neural network models.

Normalizing Flow

Estimating the density $\mathrm{p}_{\mathbf{X}}(\mathbf{x})$ of sample $\mathbf{x}$ is a classical statistical problem in various fields including computer vision and machine learning in, e.g., uncertainty estimation (Gal and Ghahramani, 2015, 2016). For tractable computation throughout the network, Bayesian adaptations are popular (Ranganath et al., 2015; Fortunato et al., 2017; Papamakarios and Murray, 2016; Kingma et al., 2015; Kendall and Gal, 2017), but these methods make assumptions on the prior distributions (e.g., exponential families).

A *normalizing flow* (Rippel and Adams, 2013; Rezende and Mohamed, 2015) first learns a function $\mathrm{f}(\cdot)$ which maps a sample $\mathbf{x}$ to a latent variable $\mathbf{z} = \mathrm{f}(\mathbf{x})$ where $\mathbf{z}$ is from a standard normal distribution $\mathbf{Z}$. Then, with a

change of variables formula, we estimate

$$p_X(\mathbf{x}) = p_Z(\mathbf{z})/|J_X|, \quad |J_X| = \left|\frac{\partial[\mathbf{x} = f^{-1}(\mathbf{z})]}{\partial \mathbf{z}}\right| \tag{6.1}$$

where $|J_X|$ is a Jacobian determinant. Thus, $f(\cdot)$ must be invertible, i.e., $\mathbf{x} = f^{-1}(\mathbf{z})$, and to use a neural network for $f(\cdot)$, a *coupling layer* structure was introduced in Real-NVP (Dinh et al., 2014, 2016) for an easy inversion and efficient $|J_X|$ computation as we describe next.

Forward map (*Fig. 6.1a*). Without loss of generality, in the context of network structures, we use an input $\mathbf{u} \in \mathbb{R}^d$ and an output $\mathbf{v} \in \mathbb{R}^d$ (i.e., $\mathbf{u} \to \mathbf{v}$). First, we split $\mathbf{u}$ into $\mathbf{u}_1 \in \mathbb{R}^{d_1}$ and $\mathbf{u}_2 \in \mathbb{R}^{d_2}$ where $d = d_1 + d_2$ (e.g., partition $\mathbf{u} \to [\mathbf{u}_1, \mathbf{u}_2]$). Then, we forward map $\mathbf{u}_1$ and $\mathbf{u}_2$ to $\mathbf{v}_1$ and $\mathbf{v}_2$ respectively:

$$\mathbf{v}_1 = \mathbf{u}_1, \quad \mathbf{v}_2 = \mathbf{u}_2 \otimes \exp(s(\mathbf{u}_1)) + r(\mathbf{u}_1) \tag{6.2}$$

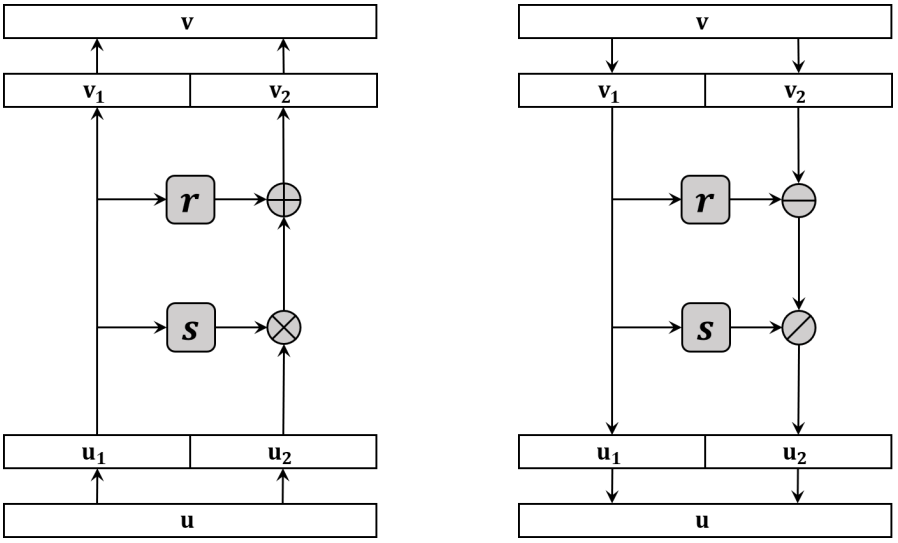
where s and r are independent functions (i.e., subnetworks), and $\otimes$ and $+$ are element-wise product and addition respectively. Then, $\mathbf{v}_1$ and $\mathbf{v}_2$ construct $\mathbf{v}$ (e.g., $[\mathbf{v}_1, \mathbf{v}_2] \to \mathbf{v}$).

Inverse map (*Fig. 6.1b*). Simple arithmetic allows an exact inverse from $\mathbf{v}$ to $\mathbf{u}$ (i.e., $\mathbf{v} \to \mathbf{u}$):

$$\mathbf{u}_1 = \mathbf{v}_1, \quad \mathbf{u}_2 = (\mathbf{v}_2 - r(\mathbf{v}_1)) \oslash \exp(s(\mathbf{v}_1)) \tag{6.3}$$

where the subnetworks s and r are *identical* to those used in the forward map in Eq. (6.2), and $\oslash$ and $-$ are element-wise division and subtraction respectively. Note that the subnetworks are *not* explicitly inverted, thus any arbitrarily complex network can be utilized.

Also, the Jacobian matrix $J_\mathbf{v} = \partial\mathbf{v}/\partial\mathbf{u}$ is triangular so its determinant $|J_\mathbf{v}|$ is just the product of the diagonal entries (i.e., $\prod_i \exp(s(\mathbf{u}_1))_i$) which is extremely easy to compute (we will discuss this further in Sec. 6.3).



(a) Forward map (Eq. (6.2)) (b) Inverse map (Eq. (6.3))

Figure 6.1: Coupling layer in normalizing flow. Note the change of operation orders: $\mathbf{u} \to \mathbf{v}$ in forward and $\mathbf{v} \to \mathbf{u}$ in inverse.

To transform the "bypassed" split $\mathbf{u}_1$ (since $\mathbf{u}_1 = \mathbf{v}_1$), a *coupling block* consisting of two complementary coupling layers stacked on top of each other with the transforming partition "swapped" (i.e., transform $\mathbf{u}_1$ in the bottom coupling layer and $\mathbf{u}_2$ in the top coupling layer) is constructed to transform both $\mathbf{u}_1$ and $\mathbf{u}_2$:

$$\begin{aligned} \mathbf{v}_1 &= \mathbf{u}_1 \otimes \exp(s_2(\mathbf{u}_2)) + r_2(\mathbf{u}_2) \\ \mathbf{v}_2 &= \mathbf{u}_2 \otimes \exp(s_1(\mathbf{v}_1)) + r_1(\mathbf{v}_1) \end{aligned} \tag{6.4}$$

and its inverse

$$\begin{aligned} \mathbf{u}_2 &= (\mathbf{v}_2 - r_1(\mathbf{v}_1)) \oslash \exp(s_1(\mathbf{v}_1)) \\ \mathbf{u}_1 &= (\mathbf{v}_1 - r_2(\mathbf{u}_2)) \oslash \exp(s_2(\mathbf{u}_2)). \end{aligned} \tag{6.5}$$

Such a series of transformations allow a more complex mapping which still comes with a chain of efficient Jacobian determinant computations, i.e., $\det(AB) = \det(A)\det(B)$ where A and B are the Jacobian matrices of two coupling layers.

Note that we have used (and will be using) $\mathbf{u}$ and $\mathbf{v}$ as generic input and output of an INN. Thus, specifically in the context of normalizing flow, by simply considering $\mathbf{u}$ and $\mathbf{v}$ to be $\mathbf{x}$ and $\mathbf{z}$ respectively, we can use a coupling layer based INN as a powerful *invertible* function $f(\cdot)$ to perform the normalizing flow described in Eq. (6.1).

6.3 Model Setup: Conditional Recurrent Flow

In this section, we describe our conditional sequence generation method called *Conditional Recurrent Flow* (CRow). We first describe a *conditional invertible neural network* (cINN) (Ardizzone et al., 2018) which is one component of our model. Then, we explain how to incorporate temporal context gating and discuss the settings where CRow can be useful.

Conditional Sample Generation

Naturally, an inverse problem can be posed as a sample generation procedure by *sampling* a latent variable $\mathbf{z}$ and inverse mapping it to $\mathbf{x} = f^{-1}(\mathbf{z})$, thus *generating* a new data $\mathbf{x}$. The most critical concern is that we cannot specifically 'choose' to generate an $\mathbf{x}$ of interest since a latent variable $\mathbf{z}$ does not provide any interpretable associations with $\mathbf{x}$.

In other words, estimating the *conditional probability* $p(\mathbf{x}|\mathbf{y})$ is desirable since it represents an underlying phenomenon of the input $\mathbf{x} \in \mathbb{R}^d$ and covariate $\mathbf{y} \in \mathbb{R}^k$ (e.g., the probability of a specific brain imaging measure $\mathbf{x}$ of interest given a diagnosis $\mathbf{y}$). In fact, when we cast this problem into a normalizing flow problem, our focus should be to construct an invertible network $f(\cdot)$ which maps a given input $\mathbf{x} \in \mathbb{R}^d$ to its corresponding covariate/label $\mathbf{y} \in \mathbb{R}^k$ and its latent variable $\mathbf{z} \in \mathbb{R}^m$ such that $[\mathbf{y}, \mathbf{z}] = f(\mathbf{x})$. The mapping must have an inverse for $\mathbf{x} = f^{-1}([\mathbf{y}, \mathbf{z}])$ to be recovered. We note that a normalizing flow problem itself does not enable a generation process since the early work on normalizing flow originally aimed to allow more complex approximation of posterior as opposed to simple the Gaussian

distribution assumption commonly found in variational inference. NiCE (Dinh et al., 2014) and Real NVP (Dinh et al., 2016) first allowed a fast transformation of the posteriors for high-dimensional features such as images, and with an invertible function, realistic sample generation was a natural functionality that has been demonstrated by recent work (Dinh et al., 2016; Kingma and Dhariwal, 2018).

Specifically, when a flow-based model jointly encodes the label and latent information (i.e., $[\mathbf{y}, \mathbf{z}] = \mathbf{v} = f(\mathbf{x})$ via Eq. (6.4)) while ensuring that $p(\mathbf{y})$ and $p(\mathbf{z})$ are independent, then the network becomes *conditionally* invertible (i.e., $\mathbf{x} = f^{-1}([\mathbf{y}, \mathbf{z}])$ conditioned on given $\mathbf{y}$). Such a network can be theoretically constructed through a bidirectional-type training (Ardizzone et al., 2018), and this allows a conditional sampling $\mathbf{x} = f^{-1}([\mathbf{y}, \mathbf{z}])$ and the posterior estimation $p(\mathbf{x}|\mathbf{y})$.

Loss functions

There are three loss functions for the bidirectional training (Ardizzone et al., 2018). Without loss of generality, let us consider a single time point which can simply be extended to multiple time points by computing these losses to each of the time points.

1. $\mathcal{L}_Z(p(\mathbf{y}, \mathbf{z}), p(\mathbf{y})p(\mathbf{z}))$: This is a loss in the forward mapping. Specifically, given a input $\mathbf{x}$, we first forward map it to $[\hat{\mathbf{y}}, \mathbf{z}] = f(\mathbf{x})$ which corresponds to $p(\hat{\mathbf{y}}, \mathbf{z})$ as our network maps both $\mathbf{y}$ and $\mathbf{z}$ with a single network. Our goal is to minimize the distance between this distribution resulting from our network ($p(\hat{\mathbf{y}}, \mathbf{z})$) to the ideal joint distribution $p(\mathbf{y})p(\mathbf{z})$. But since we may not exactly know $p(\mathbf{y})$ and $p(\mathbf{z})$, a kernel-based moment matching measure called Maximum Mean Discrepancy (MMD) (Dziugaite et al., 2015) is used which only uses the samples without explicitly requiring $p(\mathbf{y})$ and $p(\mathbf{z})$. Specifically, for each $\mathbf{x}$ and its corresponding forward map $[\hat{\mathbf{y}}, \mathbf{z}]$, we also construct its "counterpart"

sample $[\mathbf{y}_{gt}, \mathbf{z} \sim \mathbf{Z}]$ which is simply the ground truth $\mathbf{y}_{gt}$ and a random sample $\mathbf{z}$ from a standard normal $\mathbf{Z}$. In other words, we construct a set of samples representing the joint distribution $p(\mathbf{y})p(\mathbf{z})$ by empirically setting $\mathbf{y}_{gt}$ and a sample $\mathbf{z}$ from the true prior $\mathbf{Z}$ which we have been assuming. Thus, the loss is fully expressed in practice as follows:

$$\mathcal{L}_{\mathbf{Z}}(p(\mathbf{y},\mathbf{z}), p(\mathbf{y})p(\mathbf{z})) = MMD([\hat{\mathbf{y}},\mathbf{z}]_{i=1}^{N} = f(\mathbf{x}_{i=1}^{N}), [(\mathbf{y}_{gt})_{i=1}^{N}, \mathbf{z}_{i=1}^{N} \sim \mathbf{Z}]) \tag{6.6}$$

for N samples in each mini-batch. The kernel used in the MMD is an inverse multiquadratic kernel

$$k(\mathbf{x},\mathbf{x}') = \frac{\alpha}{\alpha + ||\mathbf{x}-\mathbf{x}'||_2^2}$$

where we used $\alpha = \{0.2, 0.5, 0.8, 1.0, 1.2\}$ for multiple scales of α (Ardizzone et al., 2018; Tolstikhin et al., 2017).

2. $\mathcal{L}_{\mathbf{Y}}(\mathbf{y}, \mathbf{y}_{gt})$: This is another loss in the forward mapping. Similar to typical supervised loss, it penalizes the difference between the true $\mathbf{y}_{gt}$ and the predicted $\mathbf{y}$. We used the mean squared error (MSE) function.

3. $\mathcal{L}_{\mathbf{X}}(p(\mathbf{x}), p_{\mathbf{X}})$: This is a loss in the inverse mapping (hence, the bidirectional training together with the above losses). Intuitively, this enforces the reconstructed $\mathbf{x}_{reconst}$ with known $\mathbf{y}$ and random $\mathbf{z}$ (hence a generated $\mathbf{x}$ with the estimated $p(\mathbf{x})$) to follow a likely $\mathbf{x}$ from the data with the same $\mathbf{y}$ (hence a real sample $\mathbf{x}$ from the real data $p_{\mathbf{X}}$ which in practice is from the training set). Instead of maximizing the log likelihood of $p(\mathbf{x})$ directly, this is again achieved via MMD that for a given set of $\mathbf{x}_{i=1}^{N}$ (and their $\mathbf{y}_{i=1}^{N}$), we construct a set of random samples with random $\mathbf{z}$ and the same set of $\mathbf{y}_{i=1}^{N}$ to perform the kernel-based distance measure. We use the same kernel function (and α's) as $\mathcal{L}_{\mathbf{Z}}$.

For all these losses, the ratios of the terms were all equal throughout the experiments.

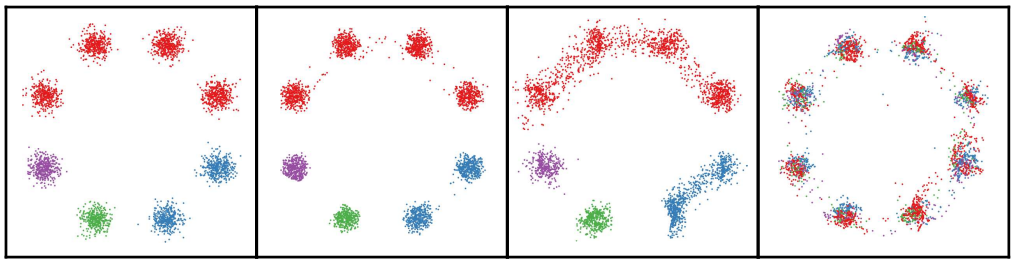


Figure 6.2: Figure 2 from (Ardizzone et al., 2018) showing the effect of each loss function. $\mathcal{L}_y$: Loss for y. $\mathcal{L}_z$: Loss for the independence of y and z. $\mathcal{L}_x$: Loss on the prior of the data, x. **Ground truth**: The ground truth distribution of the samples which are color-coded based on their labels y. **INN, all losses**: The full bi-directional training including all the losses. **INN, only** $\mathcal{L}_y + \mathcal{L}_z$: The generation result without the prior loss $\mathcal{L}_x$ showing the "bridges" between the point clusters. **INN, only** $\mathcal{L}_x$: The generation result with only the prior loss, so the labels (colors) are not considered during the generation.

In practice, $\mathbf{x}$ and $[\mathbf{y}, \mathbf{z}]$ may not be of the same dimensions. To construct a square triangular Jacobian matrix, zero-padding both $\mathbf{x}$ and $[\mathbf{y}, \mathbf{z}]$ such that their padded dimensions (i.e., the dimensions of the newly padded vectors) are the same can alleviate this issue. This also increases the intermediate subnetwork dimensions (which is equivalent to the newly padded dimension) for higher expressive power (Dinh et al., 2016; Ardizzone et al., 2018). Note that the forward mapping is essentially a prediction task that we encounter often in computer vision and machine learning, i.e., predicting $\mathbf{y} = f(\mathbf{x})$. On the other hand, the inverse process of recovering $\mathbf{x} = f^{-1}(\mathbf{y})$ may allow more interesting scientific analyses to understand the underlying relationships between $\mathbf{x}$ and $\mathbf{y}$. For instance, in the context of AD in this chapter, we try to capture and understand the underlying association between the pathology measures ($\mathbf{x}$) and the cognitive function ($\mathbf{y}$).

How to impose stable inversion?

Potential support mismatch between the data space ($\mathbf{u}$) and the latent space ($\mathbf{v}$) (e.g., when we sample a specific $\mathbf{v}$ which might not have been

mapped properly to the data, and $\mathbf{u}$ is generated from that $\mathbf{v}$) needs to be explicitly considered. This is a well-known problem in generative models (Roth et al., 2017; Tolstikhin et al., 2017; Li et al., 2017a, 2015, 2017b), and each these models often attempts to alleviate this issue in multiple ways which often depend on the type of data (e.g., generated images can be visually assessed) or the training procedure (e.g., CINN (Ardizzone et al., 2018) involves the bi-directional training). Below, we list the set of approaches that our model (and CINN models) incorporates to address this issue:

1. During the training, we impose a prior on the input side as well. This was shown to empirically stabilize the generation results using both (Ardizzone et al., 2018) and our model. Fig. 6.2 shows the effect of the prior loss $\mathcal{L}_x$ which results in cleaner generations.

2. Another small trick was used in the training where we add a small noise (e.g., independent Gaussian noise with small variance) to all the variables (i.e., $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$). Empirically, this simple data augmentation provides an additional stability to the mapping and is implicitly incorporated (usually found in the code) in many of the generative models (Dinh et al., 2016; Ardizzone et al., 2018; Kingma and Dhariwal, 2018).

3. We also describe a specific distance metric we used for all the losses involving probability distributions (e.g., $\mathcal{L}_Z$ and $\mathcal{L}_X$ that we mention in Fig. 6.2). We note that while this metric may not directly address the "outlier" generation issues, we provide details about the metric since having an accurate distance measure for probability distributions is a crucial aspect in generative models in general.

   Here, we describe a popular family of loss function that generative models often utilize to measure the distance between probabilities. Typically GANs minimize the divergence (e.g., the Jensen-Shannon divergence) between the generator and target distributions. The issue

is that they are often supported on high-dimensional manifolds, so the manifolds may not intersect, making these divergence poor choices for computing meaningful gradients (Bińkowski et al., 2018).

Acknowledging this, many of the generative models from the recent literature incorporate loss functions from a family of integral probability metrics (IPM) (Sriperumbudur et al., 2009) which better define the distance measures on probabilities. Specifically, IPM (Milgrom and Segal, 2002) can be generally formulated as

$$\mathrm{IPM}(P, Q) = \sup_{f \in \mathcal{Q}} |\mathbb{E}_P[f(X)] - \mathbb{E}_Q[f(X)]|$$

where $\mathcal{Q}$ is a class of real-valued bounded measurable functions. In other words, non-overlapping but similar distributions are measured properly based on their discrepancy (e.g., the "Earth mover" distance) in expectations over well-behaved functions .

In fact, some of the frameworks from the above related literature, in addition to the conditional INN (Ardizzone et al., 2018) and ours, incorporate maximum mean discrepancy (MMD) (Gretton et al., 2012) for a stable training. Specifically, MMD is the distance between two distributions P and Q in terms of their *mean embeddings* μ_P and μ_Q defined in a RKHS $\mathcal{H}$:

$$\mathrm{MMD}(P, Q) = ||\mu_P - \mu_Q||^2_{\mathcal{H}}.$$

In practice, we use the kernel trick to compute this over the given set of N samples $p \in P$ and M samples $q \in Q$:

$$\mathrm{MMD}(P, Q) = \frac{1}{N^2} \sum_{i=1,i'=1}^{N} k(p_i, p_{i'}) - \frac{2}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} k(p_i, q_j) + \frac{1}{M^2} \sum_{j=1,j'=1}^{M} k(q_j, q_{j'})$$

where $k(\cdot, \cdot)$ is a continuous kernel function, so $\mathrm{MMD}(P, Q)$ is zero if and only if $P = Q$.

4. With all these efforts, there is still no guarantee that the generated samples are not "out of manifold", which many studies on generative models still work towards. In some cases, the generated samples can be qualitatively assessed, which is often possible with images. In other cases where it is less intuitive to assess the quality of the generation, the estimated density information of the generated samples can be used to heuristically filter out those samples that are likely to be "out of domain".

Conditional Recurrent Flow (CRow)

The existing normalizing flow type networks cannot explicitly incorporate sequential data which are now increasingly becoming important in various applications. Successful recurrent models such as gated recurrent unit (GRU) (Chung et al., 2014; Tang et al., 2015) and Long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997; Sak et al., 2014) explicitly focus on encoding the "memory" from the past and output proper state information for accurate sequential predictions given the past. Similarly, generated sample sequences must also follow sequentially sensible patterns or trajectories resembling likely sequences by encoding appropriate *temporal information* for the subsequent time points.

To overcome these issues, we introduce our Conditional Recurrent Flow (CRow) model for conditional sequence generation. Given a sequence of input/output pairs $\{\mathbf{u}^t, \mathbf{v}^t\}$ for $t = 1, \ldots, T$ time points, modeling the relationship between the variables across time needs to also account for the temporal characteristic of the sequence. Variants of recurrent neural networks (RNN) such as GRU and LSTM have been showing success in sequential problems, but they only enable forward mapping. We are specifically interested in an *invertible network which is also recurrent* such that given a *sequence* of inputs $\{\mathbf{u}^t\}$ (i.e., features $\{\mathbf{x}^t\}$) and their *sequence* of outputs $\{\mathbf{v}^t\}$ (i.e., covariates/labels and latent information $\{\mathbf{y}^t, \mathbf{z}^t\}$), we can

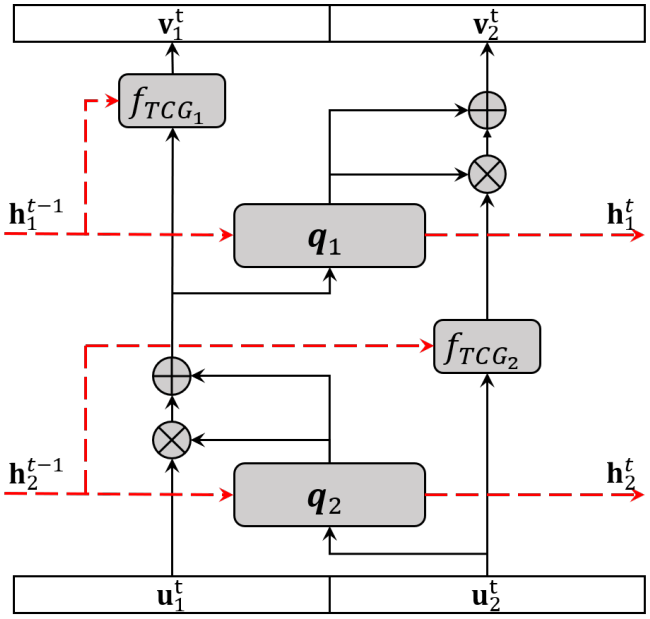


Figure 6.3: CRow architecture. Only the forward map of a single block (two coupling layers) is shown for brevity. The inverse map involves a similar order of operations (analogous to Fig. 6.1a and Fig. 6.1b).

model the invertible relationship between those sequences for posterior estimation and conditional sequence generation as illustrated in Fig. 1.8.

Without loss of generality, we can describe our model in terms of generic $\{\mathbf{u}^t\}$ and $\{\mathbf{v}^t\}$. We follow the coupling block described in Eq. (6.4) and Eq. (6.5) to setup a normalizing flow type invertible model. Then, we impose the recurrent nature on the model by allowing the model to learn and pass down a hidden state $\mathbf{h}^t$ (i.e., a temporal latent representation) to the next time point through the recurrent subnetworks. Specifically, we construct a *recurrent* subnetwork q which also contains a recurrent network (e.g., GRU) internally. This allows q to take the previous hidden state $\mathbf{h}^{t-1}$ and output the next hidden state $\mathbf{h}^t$ as $[\mathbf{q}, \mathbf{h}^t] = q(\mathbf{u}, \mathbf{h}^{t-1})$ where $\mathbf{q}$ is an element-wise transformation vector derived from $\mathbf{u}$ analogous to the output of a subnetwork $s(\mathbf{u})$ in Eq. (6.2). In previous coupling layers (i.e., Eq. (6.2)), two transformation vectors $\mathbf{s} = s(\cdot)$ and $\mathbf{r} = r(\cdot)$ were explicitly

computed from two subnetworks for each layer. For CRow, we follow the structure of Glow (Kingma and Dhariwal, 2018) which computes a single vector $\mathbf{q} = q(\cdot)$ and splits it as $[\mathbf{s}, \mathbf{r}] = \mathbf{q}$. This allows us to use a single hidden state while concurrently learning $[\mathbf{s}, \mathbf{r}]$ which we denote as $\mathbf{s} = q_s(\cdot)$ and $\mathbf{r} = q_r(\cdot)$ to indicate the individual vectors. Thus, at each t with given $[\mathbf{u}_1^t, \mathbf{u}_2^t] = \mathbf{u}^t$ and $[\mathbf{v}_1^t, \mathbf{v}_2^t] = \mathbf{v}^t$,

$$\begin{aligned} \mathbf{v}_1^t &= \mathbf{u}_1^t \otimes \exp(q_{s_2}(\mathbf{u}_2^t, \mathbf{h}_2^{t-1})) + q_{r_2}(\mathbf{u}_2^t, \mathbf{h}_2^{t-1}) \\ \mathbf{v}_2^t &= \mathbf{u}_2^t \otimes \exp(q_{s_1}(\mathbf{v}_1^t, \mathbf{h}_1^{t-1})) + q_{r_1}(\mathbf{v}_1^t, \mathbf{h}_1^{t-1}) \end{aligned} \tag{6.7}$$

and the inverse is

$$\begin{aligned} \mathbf{u}_2^t &= (\mathbf{v}_2^t - q_{r_1}(\mathbf{v}_1^t, \mathbf{h}_1^{t-1})) \oslash \exp(q_{s_1}(\mathbf{v}_1^t, \mathbf{h}_1^{t-1})) \\ \mathbf{u}_1^t &= (\mathbf{v}_1^t - q_{r_2}(\mathbf{u}_2^t, \mathbf{h}_2^{t-1})) \oslash \exp(q_{s_2}(\mathbf{u}_2^t, \mathbf{h}_2^{t-1})). \end{aligned} \tag{6.8}$$

Note that the hidden states $\mathbf{h}_1^t$ and $\mathbf{h}_2^t$ generated from the recurrent network of the subnetworks are *explicitly* used within the subnetwork architecture (i.e., inputs to additional fully connected layers) and also passed to their corresponding recurrent network in the next time point as in Fig. 6.3. Again, similar to the hidden state learned by SP-GRU from Chapter 5, the hidden state which CRow generates and explicitly utilizes is exactly the kind of temporal latent representation that we sought to construct in order to encode the temporal patterns of complex sequences.

Temporal Context Gating (TCG)

A standard (single) coupling layer transforms only a part of the input (i.e., $\mathbf{u}_1$ in Eq. (6.2)) by design which results in the determinant of a triangular Jacobian matrix $J_\mathbf{v}$:

$$|J_\mathbf{v}| = \left|\frac{\partial \mathbf{v}}{\partial \mathbf{u}}\right| = \begin{vmatrix} \frac{\partial \mathbf{v}_1}{\partial \mathbf{u}_1} & \frac{\partial \mathbf{v}_1}{\partial \mathbf{u}_2} \\ \frac{\partial \mathbf{v}_2}{\partial \mathbf{u}_1} & \frac{\partial \mathbf{v}_2}{\partial \mathbf{u}_2} \end{vmatrix} = \begin{vmatrix} I & 0 \\ \frac{\partial \mathbf{v}_2}{\partial \mathbf{u}_1} & \mathrm{diag}(\exp s(\mathbf{u}_1)) \end{vmatrix} \tag{6.9}$$

thus $|J_{\mathbf{v}}| = \exp(\sum_i (s(\mathbf{u}_1))_i)$. This is a result from Eq. (6.2): (1) the element-wise operations on $\mathbf{u}_2$ for the diagonal submatrix of partial derivatives $\partial \mathbf{v}_2 / \partial \mathbf{u}_2 = \mathrm{diag}(\exp s(\mathbf{u}_1))$, (2) the bypassing of $\mathbf{u}_1 = \mathbf{v}_1$ for $\partial \mathbf{v}_1 / \partial \mathbf{u}_1 = I$, and (3) $\partial \mathbf{v}_1 / \partial \mathbf{u}_2 = 0$. Ideally, transforming $\mathbf{u}_1$ would be beneficial. However, this is explicitly avoided in the coupling layer design since this should *not* involve $\mathbf{u}_1$ or $\mathbf{u}_2$ directly; otherwise, $J_{\mathbf{v}}$ will not be triangular.

The Jacobian determinants of the subsequent coupling layers can be computed consecutively using the output of the previous coupling layer as the input to the current coupling layers. In other words, for a series of composited formulations $f = f_1 \circ f_2 \circ \cdots \circ f_N$ where each f_i is a coupling layer operation, then $\det(f)$ is

$$\det(f) = \det(f_1 \circ f_2 \circ \cdots \circ f_N) = \det(f_1) \det(f_2) \cdots \det(f_N) = \prod_{i=1}^{N} \det(f_i). \tag{6.10}$$

This allows an easy computation of the full Jacobian determinant across the layers regardless of the number of coupling layer operations because we do not have to perform a series of matrix multiplications across the Jacobian matrices which is computationally more demanding than simply summing over the products of the diagonal entries.

Using $\mathbf{h}^t$ *in CRow.* In the case of CRow, it incorporates a hidden state $\mathbf{h}^{t-1}$ from the previous time point which is *neither* $\mathbf{u}$ nor $\mathbf{v}$. This is our *temporal information* which adjusts the mapping function $f(\cdot)$ to allow more accurate mapping depending on the previous time points of the sequence which is crucial for sequential modeling.

Specifically, we incorporate a *temporal context gating* $f_{\mathrm{TCG}}(\alpha^t, \mathbf{h}^{t-1})$ using the temporal information $\mathbf{h}^{t-1}$ on a given input α^t at t as follows:

$$\begin{aligned} f_{\mathrm{TCG}}(\alpha^t, \mathbf{h}^{t-1}) &= \alpha^t \otimes \mathrm{cgate}(\mathbf{h}^{t-1}) \quad \text{(forward)} \\ f^{-1}_{\mathrm{TCG}}(\alpha^t, \mathbf{h}^{t-1}) &= \alpha^t \oslash \mathrm{cgate}(\mathbf{h}^{t-1}) \quad \text{(inverse)} \end{aligned} \tag{6.11}$$

where $\mathtt{cgate}(\mathbf{h}^{t-1})$ can be any learnable function/network with a sigmoid function at the end. This is analogous to the context gating (Miech et al., 2017) in video analysis which scales the input α^t (since $\mathtt{cgate}(\mathbf{h}^{t-1}) \in (0,1)$) based on some relevant context, which in our setup is the temporal information $\mathbf{h}^{t-1}$.

Preserving the Jacobian structure. In the context of $|J_{\mathbf{v}}|$ computation in Eq. (6.9), we perform $f_{TCG}(\mathbf{u}_1, \mathbf{h}^{t-1}) = \mathbf{u}_1 \otimes \mathtt{cgate}(\mathbf{h}^{t-1})$ (w.l.o.g., we omit t for $\mathbf{u}$ and $\mathbf{v}$). Importantly, we observe that this 'auxiliary' variable $\mathbf{h}^{t-1}$ could safely be used to transform $\mathbf{u}_1$ *without* altering the triangular structure of the Jacobian matrix for the following two advantages: (1) we still perform an element-wise operation $\mathbf{u}_1 \otimes \mathtt{cgate}(\mathbf{h}^{t-1})$ resulting in a diagonal submatrix for $\partial \mathbf{v}_1 / \partial \mathbf{u}_1$, and (2) $\partial \mathbf{v}_1 / \partial \mathbf{u}_2$ is still 0 since $\mathbf{u}_2$ is not involved in $f_{TCG}(\mathbf{u}_1, \mathbf{h}^{t-1})$. If the resulting Jacobian matrix were not triangular, then the determinant computation would be much more complex than the simple multiplication of the diagonal elements of a triangular Jacobian matrix which would be critical in neural networks with high intermediate feature dimensions. Thus, we now have

$$|J_{\mathbf{v}}| = \begin{vmatrix} \frac{\partial \mathbf{v}_1}{\partial \mathbf{u}_1} & \frac{\partial \mathbf{v}_1}{\partial \mathbf{u}_2} \\ \frac{\partial \mathbf{v}_2}{\partial \mathbf{u}_1} & \frac{\partial \mathbf{v}_2}{\partial \mathbf{u}_2} \end{vmatrix} = \begin{vmatrix} \mathtt{diag}(\mathtt{cgate}(\mathbf{h}^{t-1})) & 0 \\ \frac{\partial \mathbf{v}_2}{\partial \mathbf{u}_1} & \mathtt{diag}(\exp s(\mathbf{u}_1)) \end{vmatrix} \tag{6.12}$$

where $|J_{\mathbf{v}}| = [\prod_j \mathtt{cgate}(\mathbf{h}^{t-1})_j] * [\exp(\sum_i (s(\mathbf{u}_1))_i)]$.

As seen in Fig. 6.3, we place f_{TCG} to transform the "bypassing" split (non-transforming partition) of each *layer* of a block (i.e., the "bypassing" partition $\mathbf{u}_2^t$ gets transformed by f_{TCG_2}). We specifically chose a gating mechanism for conservative adjustments so that the original information is preserved to a large degree through simple but learnable 'weighting'. The full forward and inverse steps involving f_{TCG} can easily be formulated by following Eq. (6.7) and Eq. (6.8) while respecting the order of operations seen in Fig. 6.3.

How do we use CRow?

In essence, CRow aims to model an invertible mapping $[\{\mathbf{y}^t\},\{\mathbf{z}^t\}] = f(\{\mathbf{x}^t\})$ between sequential/longitudinal measures $\{\mathbf{x}^t\}$ and their corresponding observations $\{\mathbf{y}^t\}$ with $\{\mathbf{z}^t\}$ encoding the latent information across $t = 1,\ldots,T$ time points. Once we train $f(\cdot)$, we can perform the following exemplary tasks:

(1) Conditional sequence generation: Given a series of observations of interest $\{\mathbf{y}^t\}$, we can sample $\{\mathbf{z}^t\}$ (each independently from a standard normal distribution) to generate $\{\mathbf{x}^t\} = f^{-1}([\{\mathbf{y}^t\},\{\mathbf{z}^t\}])$. The advantage comes from how $\{\mathbf{y}^t\}$ can be flexibly constructed (either seen or unseen from the data) such as an arbitrary disease progression over time (see Fig. 1.8). Then, we randomly generate corresponding measures $\{\mathbf{x}^t\}$ to observe the corresponding longitudinal measures for both quantitative and qualitative analyses. Since the model is recurrent, the sequence length can be extended beyond the training data to model future trajectories. In fact, in Chapter 7, we will see how CRow can be used to estimate the trajectories beyond the observed time points for characterizing the longitudinal pattern of sequential brain imaging measures.

(2) Sequential density estimation: Conversely, given $\{\mathbf{x}^t\}$, we can predict $\{\mathbf{y}^t\}$, and more importantly, estimate the density $p_X(\{\mathbf{x}^t\})$ at each t. When $\{\mathbf{x}^t\}$ is generated from $\{\mathbf{y}^t\}$, the estimated density can indicate the 'integrity' of the generated sample (i.e., low p_X implies that the sequence is perhaps less common with respect to $\{\mathbf{y}^t\}$).

6.4 Experiments

We validate our framework in both a qualitative and quantitative manner with two sets of experiments: (1) two image sequence datasets and (2) a neuroimaging study. We used NVIDIA 1080 Ti GPU to train all the models.

ADAM optimizer with $\alpha = 0.9$ and $\beta = 0.999$ and the initial learning rate of 0.0005 was used.

Conditional Moving MNIST Generation

Moving Digit MNIST

We first test our model on a controlled Moving Digit MNIST dataset (Srivastava et al., 2015) of image sequences showing a hand-written digit from 0 to 9 moving in a path and bouncing off the boundary. This experiment qualitatively shows that the images in a generated sequence with specific conditions (i.e., image labels) are consistent across the sequence. Here, we specifically chose two digits (e.g., 0 and 1) to construct ~13K controlled sequences of frame length $T = 6$ where each frame of a sequence is an image of size 20 by 20 (vectorized as $\mathbf{x}^t \in \mathbb{R}^{400}$) and has a one-hot vector $\mathbf{y}^t \in \mathbb{R}^2$ of digit label at t indicating one of the two possible digits. Thus, the condition $\mathbf{y}^t$ was in $\{0, 1\}^2$ (onehot vector for 2 classes case) and the latent variable $\mathbf{z}^t$ was in $\mathbb{R}^8$ which was chosen by us. For a smaller sized $\mathbf{z}^t$, the densities could not be accurately captured. On the other hand, for a larger sized $\mathbf{z}^t$, the latent information could potentially "memorize" the input to output mapping which is also undesirable. Both input and output were zero-padded to 512 dimensions.

Training. Our model consists of three coupling blocks, each block shown in Fig. 6.3, where each subnetwork q contains one GRU cell and three layers of residual fully connected networks with ReLU activation. For each TCG (f_{TCG} in Fig. 6.3, Eq. (6.11)), the network $\mathtt{cgate}(\cdot)$ is a single fully connected network with sigmoid activation. We split $\mathbf{x}$ (i.e., $\mathbf{u}$ without zero padding) into $[\mathbf{x}_1, \mathbf{x}_2]$ ($\mathbf{x}_1$ into $\mathbf{u}_1$ and $\mathbf{x}_2$ into $\mathbf{u}_2$) in a "checkerboard" pattern. In other words, given an image $\mathbf{x}$, the first half $\mathbf{x}_1$ consists of the pixels that are not directly adjacent to each other (i.e., black squares in a checkerboard) and the remaining pixels (which are also not directly adjacent to each other

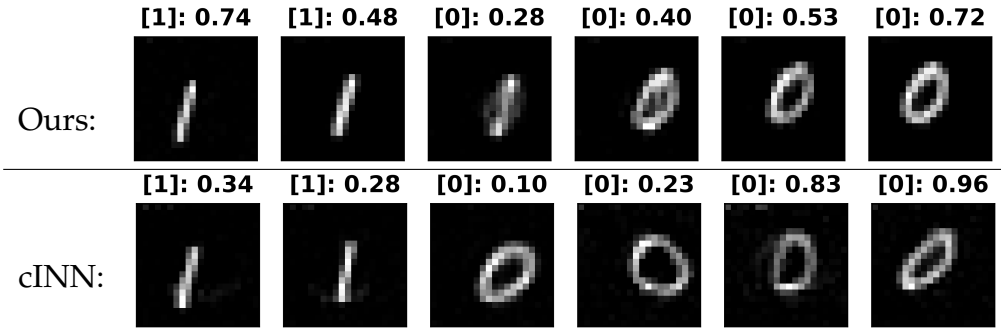


Figure 6.4: Examples of generated sequences given the changing condition 1→1→0→0→0→0 (top of each frame, **[digit label]: density**). Ours shows smooth transition while cINN shows temporally drastic transition.

like the white squares in a checkerboard) are assigned to $\mathbf{x}_2$. This splitting scheme preserves the overall geometric structure of the image as much as possible in a simplistic manner as did Real-NVP (Dinh et al., 2016). Models were trained on $T = 6$ time points, but further time points data can be generated since our model is recurrent. Each training sequence has a digit label sequence $\{\mathbf{y}^t\}$ for $t = 1, \ldots, 6$ where all $\mathbf{y}^t$ are "identical" in each sequence since the the same digit is shown throughout the sequence. We used 3 coupling blocks (2 coupling layers in each block) where the input/output dimensions (i.e., $\mathbf{u}$ and $\mathbf{v}$ dimensions) are 512 (thus, zero padding of length 112 is needed for $\mathbf{x}^t$ and 502 is needed for $[\mathbf{y}^t, \mathbf{z}^t]$). Each subnetwork q then has the input/output dimensions of 256 where it first starts with a GRU (256 input, 256 output, 256 hidden) followed by 3 residual layers (fully connected layers with ReLU non-linearity, all of 256 input and 256 output). For details about the GRU structure, refer to Sec. 5.2 in Chapter 5.

Generation. Now, we want to generate sequences showing digits that gradually transform (e.g., changing from 1 to 0). We first specified sequential conditions (i.e., digit label) that change midway through the sequence (e.g., $\{\mathbf{y}^t\}$ sequence indicating digit labels 1→1→0→0→0→0). Then, we generated the corresponding sequences $\{\mathbf{x}^t\}$ and visually check

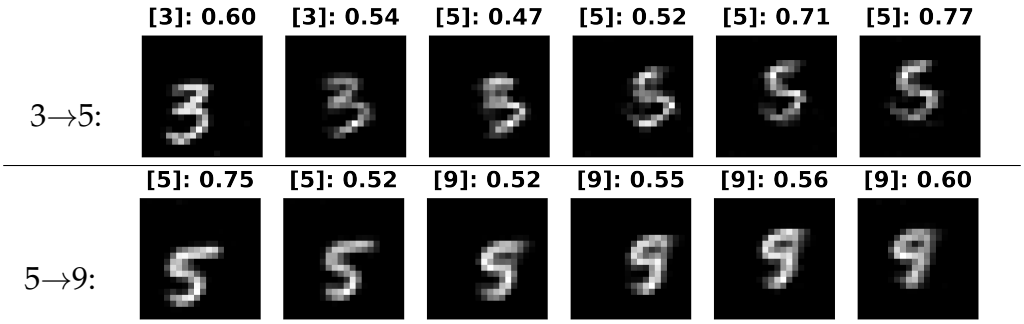


Figure 6.5: Examples of generated sequences using CRow.

if the changes across the frames look natural. Note that we trained *only* on the image sequences with consistent digit labels (e.g., only 0 or only 1). One demonstrative result is shown in Fig. 6.4 where we compare the generated image sequences with condition (i.e., digit label) changing from 1 to 0. Our result at the top of Fig. 6.4 shows a gradual transition while the cINN result does not show such temporally smooth and consistent behavior.

Density estimation. Our model quantifies its output confidence in a form of density (i.e., likelihood) shown at the top of each generated images in Fig. 6.4. Not only our model transforms the sequence generation based on the sequential condition, but it also outputs a lower density at the frame showing the most drastic transformation as such patterns were not observed during the training, i.e., the likelihood decreases when the condition changes and then increases as the sequence progresses. This means that our model simultaneously shows the conditional generation ability and estimates the output's relative density with respect to the observed training data. More examples are shown in Fig. 6.5.

Moving Fashion MNIST

We also tested our model on a more challenging dataset called Moving Fashion MNIST (Xiao et al., 2017) of moving apparel image sequences. The Moving Fashion MNIST dataset (Xiao et al., 2017) has very similar

dataset specifications as the Moving Digit MNIST dataset (Srivastava et al., 2015): (1) the original gray-scale image is of size 28×28, (2) there are 10 classes, and (3) the training set has 6K samples of each class (total of 60K). Thus, the training and testing pipelines for both datasets were identical in our experiments. In Table 6.1, we show the list of the apparel types in the Moving Fashion MNIST dataset. The same models and training

Label Index	**Apparel Type**
0	T-shirt/Top
1	Trouser
2	Pullover
3	Dress
4	Long sleeve/Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

Table 6.1: Moving Fashion MNIST apparel types.

setups were used to generate the transforming sequences in a similar manner. In Fig. 6.6, we show the examples of various apparels successfully transforming to other types while moving. Compared to Moving Digit MNIST, capturing the smooth transformations of these apparel images are more challenging as the apparel shapes vary more in terms of their shapes and sizes.

Longitudinal Neuroimaging Analysis

In this neuroimaging experiment, we evaluate if our conditionally generated samples actually exhibit statistically robust and clinically sound characteristics when trained with a longitudinal Alzheimer's disease (AD) brain imaging dataset. We generated a *sufficient* number of longitudinal brain imaging measures (i.e., $\{\mathbf{x}^t\}$) conditioned on various covariates (i.e.,

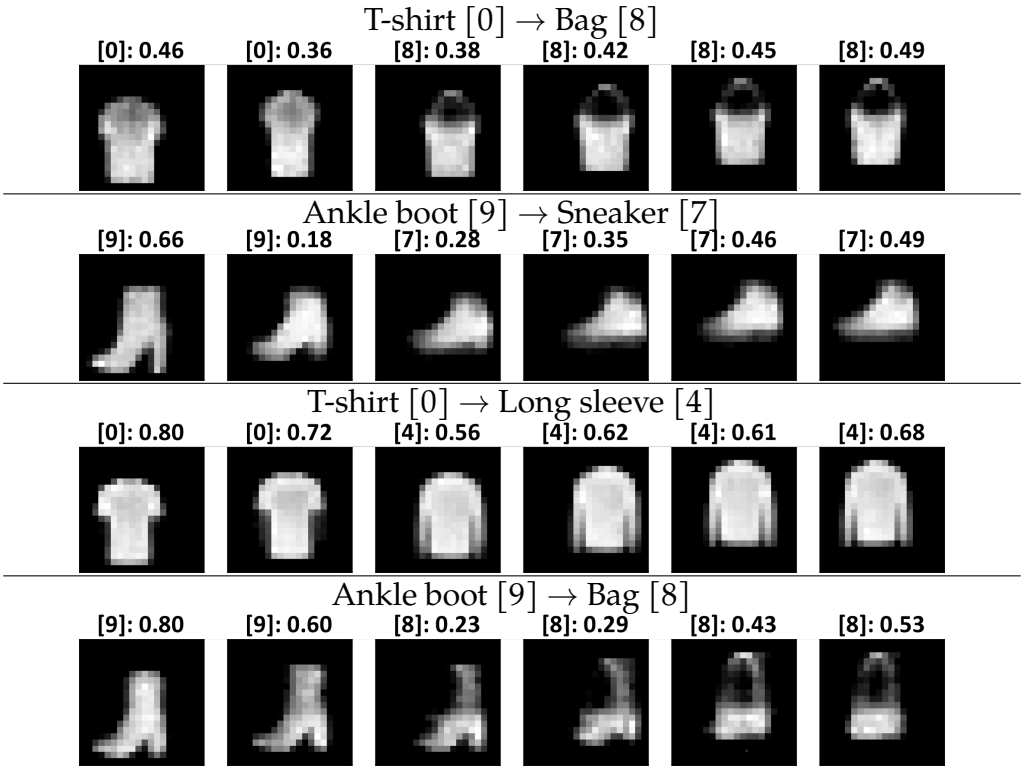


Figure 6.6: Examples of generated Moving Fashion MNIST sequences using CRow (apparel type [label index]).

labels $\{\mathbf{y}^t\}$) associated with the AD progression (e.g., cognition). Thus, the generated brain imaging sequences should show the pathology progression consistent with the covariate progression (see Fig. 1.8 and Fig. 6.8 for illustrations). We then performed a statistical group analysis (i.e., healthy vs. disease progressions) to detect disease related features from the imaging measures. In the end, we expected that the brain regions of interests (ROIs) identified by the statistical group analysis are consistent with other AD literature with statistically stronger signal (i.e., lower p-value) than the results based only on the original training data.

TADPOLE Dataset

The Alzheimer's Disease Neuroimaging Initiative (ADNI) database (`adni.loni.usc.edu`) is one of the largest and still growing neuroimaging databases. Originated from ADNI, we use a longitudinal neuroimaging dataset called The Alzheimer's Disease Prediction of Longitudinal Evolution (TADPOLE) (Marinescu et al., 2018) which actually consists of multiple datasets, each serving a different purpose with respect to the challenge itself. For our experiments, we used D1 and D2: (i) D1 is the standard training set consisting of individuals with at least two separate visits across the three phases of the ADNI study (ADNI1, ADNI GO and ADNI2) and (ii) D2 is the longitudinal prediction set which have the rollovers (i.e., subjects from D1 with further visits) for the purpose of forecasting tasks. In our setup, we simply treated the subjects in D1 and D2 without distinctions to obtain the most number of subjects with (i) 3 time points with (ii) AV45 measures for all 3 time points and (iii) covariates of interests at each time point. For this experiment, we specifically used $N = 276$ participants with $T = 3$ time points.

Input. For the longitudinal brain imaging sequence $\{\mathbf{x}^t\}$, we chose Florbetapir (AV45) Positron Emission Tomography (PET) scan measuring the level of *amyloid-beta* deposited in brain which has been a known type of pathology associated with Alzheimer's disease (Wong et al., 2010; Joshi et al., 2012). The AV45 images were registered to a common brain template (MNI152) to derive the gray matter regions of interests (82 Desikan atlas ROIs (Desikan et al., 2006)). Thus, each of the 82 ROIs ($\mathbf{x}^t \in \mathbb{R}^{82}$) holds an average Standard Uptake Value Ratio (SUVR) measure of AV45 where high AV45 implies more amyloid pathology in that region. The Desikan ROIs are illustrated in Fig. 6.7. The colors distinguish different ROIs (not reflecting any measures).

Condition. For the corresponding labels $\{\mathbf{y}^t\}$ for longitudinal conditions, we chose five covariates known to be tied to the AD progression (normal

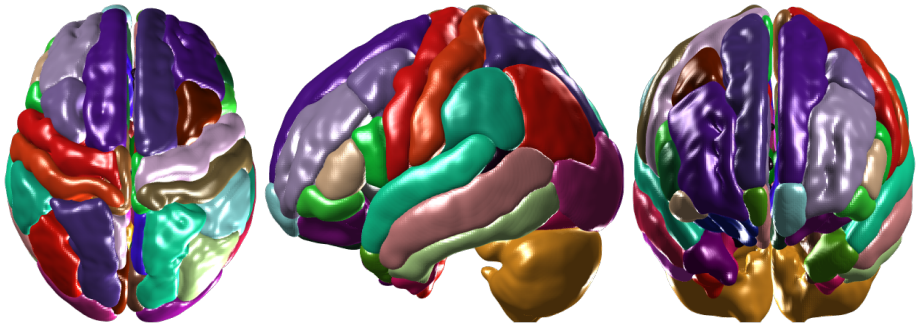


Figure 6.7: Desikan ROIs. Different colors encode different ROIs. Top row: (from left) top, left and front views. Bottom row: diagonal views.

to impaired range in square brackets): **(1)** *Diagnosis*: Normal/Control (CN), Mild Cognitive Impairment (MCI), and Alzheimer's Disease (AD) [CN→MCI→AD]. **(2)** *ADAS13*: Alzheimer's Disease Assessment Scale [0→85]. **(3)** *MMSE:* Mini Mental State Exam [0→30]. **(4)** *RAVLT-I:* Rey Auditory Verbal Learning Test - Immediate [0→75]. **(5)** *CDR:* Clinical Dementia Rating [0→18]. These assessments will *condition* the disease *progression* of the samples. Full documents are available on `http://adni.loni.usc.edu/methods/documents` (e.g., ADNI Procedures Manual). The condition $\mathbf{y}^t$ was in $\{0,1\}^3$ for *Diagnosis* (onehot vector over three possible diagnosis categories) and in $\mathbb{R}^1$ for other continuous covariates. The latent variable $\mathbf{z}^t$ was in $\mathbb{R}^4$. Both input and output were zero-padded to be of size 150. The remaining setup is exactly the same as the Moving MNIST setup.

Analysis

We performed a statistical group analysis on each condition $\{\mathbf{y}^t\}$ independently with the following pipeline: *(1) Training:* First, we trained our model (the same subnetwork as Sec. 6.4) using the sequences of SUVR in 82 ROIs for $\{\mathbf{x}^t\}$ and the covariate ('label') sequences for $\{\mathbf{y}^t\}$. *(2) Conditional longitudinal sample generation:* Then, we generated longitudinal samples $\{\hat{\mathbf{x}}^t\}$ conditioned on two distinct longitudinal conditions: *Control* (healthy

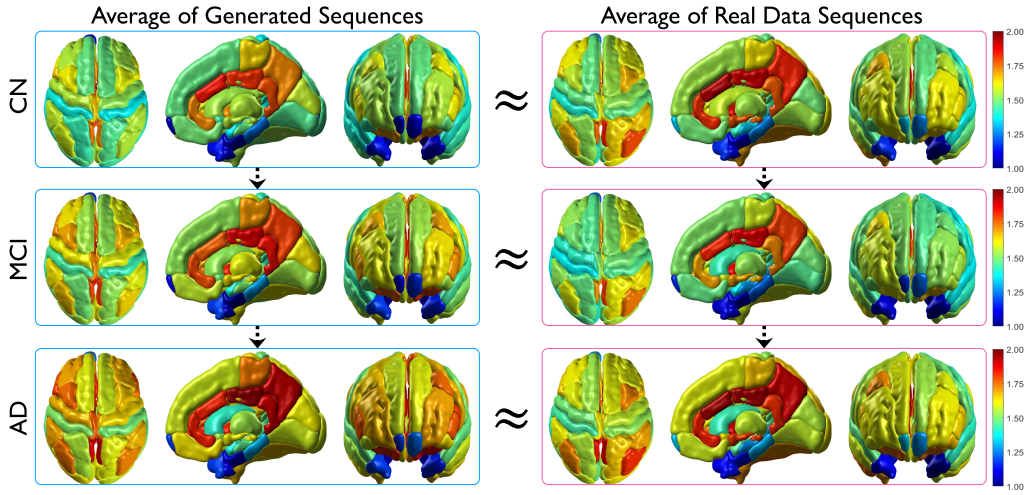


Figure 6.8: Generated sequences vs. real data sequences comparison for CN (top)→MCI (middle)→AD (bottom). Each blue/pink frame has top, side (interior of right hemisphere), and front views. **Left (blue frames):** The average of the 100 generated sequences conditioned on CN→MCI→AD. **Right (pink frames):** The average of the real samples with CN→MCI→AD in the dataset. Red/blue indicate high/low AV45. ROIs are expected to turn more red as CN→MCI→AD. The generated samples show magnitudes and sequential patterns similar (≈) to those of the real samples from the training data.

covariate sequence) versus *Progression* (worsening covariate sequence). Specifically, for each condition (e.g., Diagnosis), we generate N_1 samples of Control (e.g., $\{\hat{\mathbf{x}}_1^t\}_{i=1}^{N_1}$ conditioned on $\{\mathbf{y}^t\}_=$ CN→CN→CN) and N_2 samples of Progression ($\{\hat{\mathbf{x}}_2^t\}_{i=1}^{N_2}$ conditioned on $\{\mathbf{y}^t\}$ = CN→MCI→AD). Then, we perform a two sample t-test at $t = 3$ for each of 82 ROIs between $\{\hat{\mathbf{x}}_1^3\}_{i=1}^{N_1}$ and $\{\hat{\mathbf{x}}_2^3\}_{i=1}^{N_2}$ groups, and derive p-values to tell whether the pathology levels between the groups significantly differ in those ROIs.

Result 1: Control vs. Progression (Table 6.2, Top row block)

We set the longitudinal conditions for each covariate based on its associated to healthy progression (e.g., low ADAS13 throughout) and disease progression (e.g., high ADAS13 related to eventual AD onset). We generated $N_1 = 100$ and $N_2 = 100$ samples for each group respectively. Then, we performed the above statistical group difference analysis under 4 se-

	# of Statistically Significant ROIs (# of ROIs after type-I error correction)				
Covariates	Diagnosis	ADAS13	MMSE	RAVLT-I	CDR-SB
Control	CN→CN→CN	10→10→10	30→30→30	70→70→70	0→0→0
Progression	CN→MCI→AD	10→20→30	30→26→22	70→50→30	0→5→10
cINN	11 (4)	5 (2)	5 (0)	3 (0)	7 (0)
Ours	25 (11)	24 (12)	19 (**2**)	15 (**2**)	18 (7)
Ours + TCG	**28** (**12**)	**32** (**14**)	**31** (**2**)	**19** (**2**)	**25** (**9**)
Control	CN→CN→CN	10→10→10	30→30→30	70→70→70	0→0→0
Early-progression	CN→MCI→MCI	10→13→16	30→28→26	70→60→50	0→2→4
cINN	2 (0)	2 (2)	2 (0)	0 (0)	1 (0)
Ours	**6** (2)	6 (4)	11 (**4**)	**5** (**1**)	2 (0)
Ours + TCG	**6** (**4**)	**8** (**5**)	**12** (**4**)	**5** (**1**)	**5** (**1**)

Table 6.2: Number of ROIs identified by statistical group analysis using the generated measures with respect to various covariates associated with AD with the significance level of $\alpha = 0.01$ (type-I error controlled result shown in parenthesis). Each column represents sequences of disease progression represented by diagnosis or test scores. *CRow* considers the progression sequences while cINN generates cross-sectional data in different conditions. In all cases, using *CRow* with TCG yields the most number of statistically significant ROIs.

tups: (**1**) Raw training data, (**2**) cINN (Ardizzone et al., 2018), (**3**) Our model without TCG, and (**4**) Our model + TCG. With the raw data, the sample sizes of the desirable longitudinal conditions were extremely small for all setups, so no statistical significance was found after type-I error control. With cINN, we occasionally found few significant ROIs, but the non-sequential samples with only $t = 3$ could not generate realistic samples. With CRow (without TCG) we consistently found significant ROIs. Further, CRow + TCG detected the most number of ROIs (the ROIs for Diagnosis shown in Fig. 6.9) which include many AD-specific regions reported in the aging literature such as hippocampus and amygdala (Jin et al., 2004; Joshi et al., 2012).

Result 2: Control vs. Early-progression (Table 6.2, Bottom row block)

We setup a more challenging task where we generate samples which resemble the subjects that show a slower progression of the disease (i.e., lower rate of covariate change over time). Such case is especially important in AD when early detection leads to effective prevention. With $N_1 = 100$ and $N_2 = 100$ samples, no significant ROIs were found in all models. To

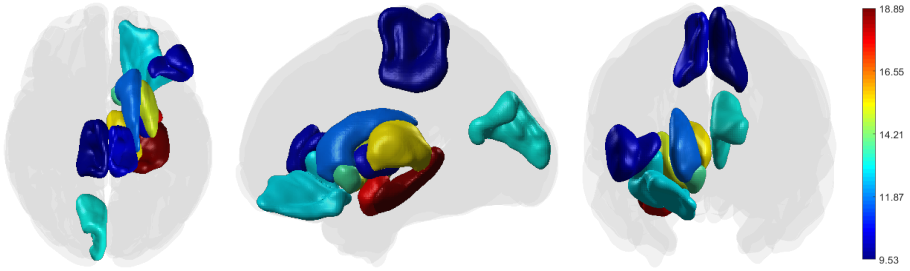


Figure 6.9: 12 significant ROIs found between two Diagnosis groups (CN→CN→CN vs. CN→MCI→AD) at t = 3 using our model under 'Diagnosis' in Table 6.2. The colors denote the -log p-value. AD-related ROIs such as hippocampus, putamen, caudate, and amygdala are included.

	ROI	p-value	
		Real	CRow
Diagnosis	Left Amygdala	5.51E-03	**1.18E-06**
	Left Putamen	7.38E-03	**3.99E-05**
ADAS13	Left Inferior Temporal	3.34E-03	**7.93E-04**
	Left Middle Temporal	6.83E-03	**2.02E-03**
MMSE	Left Superior Parietal	7.13E-03	**1.52E-05**
	Left Supramarginal	6.75E-03	**8.20E-08**
RAVLT-I	Left Paracentral	9.16E-03	**8.09E-05**
CDR-SB	Left Hippocampus	4.01E-03	**3.36E-06**

Table 6.3: *p*-values in ROIs improve (get lower) with the sequences generated by CRow with increased sample size over using real sequence data.

improve the sensitivity, we generated $N_1 = 150$ and $N_2 = 150$ samples in all models and found several significant ROIs *only* with CRow related to an *early* AD progression such as hippocampus (Fox et al., 1996; Johnson et al., 2014b).

Statistical Advantages

By generating realistic samples with CRow, we achieve the following advantages: **(1)** Increasing sample size makes the hypothesis test more sensitive and robust – rejecting the null when it is indeed false – leading to a lower type-II error. **(2)** Also, we do *not* simply detect spurious signifi-

cant ROIs because (i) we control for type-I error via the *most conservative* Bonferroni multiple testing correction, and (ii) we additionally improve the statistical power of detecting the *true effects* (i.e., significant ROIs) that *at least* need to be detected with the raw data only. In Table 6.3, we show that the significant ROIs identified with the real data only are also detected through our framework with *improved* p-values from the Control vs. Progression experiment. This suggests that we take advantage of CRow in a statistically meaningful manner *without neglecting the true signals* from the important AD-specific ROIs (Fox et al., 1996; Ossenkoppele et al., 2012). To obtain similar improved results with real data, one would have to spend *substantial resources and time* to recruit more participants and acquire their images.

Generation assessments

In Fig. 6.8, we see the generated samples (Left) through CN→MCI→AD in three views of the ROIs and compare them to the real training samples (Right). We observe that the generated samples have similar AV45 loads across the ROIs, and more importantly, the progression pattern across the progression (i.e., ROIs turning more red indicating amyloid accumulations in those ROIs) follows that of the real sequence as well. We also quantified the similarities between the generated and real data sequences by computing the effect size (Cohen's d (Cohen, 2013)) which measures the difference between the two distributions (Table 6.4) showing that CRow generates the most realistic sequences.

Scientific Remarks

Throughout our analyses, the significant ROIs that we found such as amygdala, putamen, temporal regions, hippocampus (e.g., shown in Fig. 6.9) and many others were also reported to be the AD-specific regions in the aging field (Fox et al., 1996; Jin et al., 2004; Johnson et al., 2014b; Ossenkop-

	Gen. vs. Real of Progressions					Gen. vs. Real of Early-progressions				
Covariates	Diagnosis	ADAS	MMSE	RAVLT	CDR	Diagnosis	ADAS	MMSE	RAVLT	CDR
cINN	1.26	1.60	1.15	1.89	1.55	1.07	1.50	0.95	1.84	1.45
Ours	0.42	0.56	0.35	0.71	0.65	0.36	0.56	0.30	0.61	0.63
Ours+TCG	**0.28**	**0.39**	**0.17**	**0.59**	**0.38**	**0.23**	**0.52**	**0.09**	**0.54**	**0.50**

Table 6.4: Difference between the generated sequences and the real sequences at $t = 3$. Lower the effect size (Cohen's d), smaller the difference between the comparing distributions. In all settings, *CRow* with TCG generates the most realistic sequences with the smallest effect sizes.

pele et al., 2012; Villemagne et al., 2013). This implies that the generated longitudinal sequences consistently follow the underlying distribution of the real data which we may *not* have been able to make use of otherwise.

6.5 Summary

In this chapter, we studied the problem of generative models using neural networks that account for the progressive behavior of longitudinal data sequences. By developing a novel architecture of an invertible neural network that incorporates recurrent subnetworks and temporal context gating to pass down the temporal information across the sequence generation, we enabled a neural network to "learn" the conditional distribution of training data in a latent space and generate a sequence of samples with a realistic progressive behavior according to the given conditions. We demonstrated experimental results using three datasets (2 moving videos and 1 neuroimaging) to validate longitudinal progression in sequentially generated samples. Also, in neuroimaging applications which often suffer from small sample sizes, we showed that our model can generate realistic samples for statistically robust results.

Interestingly, the work in this chapter can be quite versatile such that it can be generalized to the problems we tackled in the previous chapters. For instance, if we use both the age and covariate sequences as the condition ($\mathbf{y}$) and capture their relationships to the brain connectivity ($\mathbf{x}$),

we can generate sequences of **x** which account for both the cross-sectional progression via changing the covariate sequences and the longitudinal progression via changing the age which are exactly the types of sequences we tried to characterize in Chapter 4. Also, we could naturally use the density estimation outcome of the sequential predictions that CRow makes as an uncertainty measure which SP-GRU from Chapter 5 focused on. In fact, in the next chapter, we will see how another functionality of CRow could extensively be used as a robust sequential prediction model to accurately predict the pathology trajectory to understand the early pathological process of Alzheimer's disease.

7 PREDICTING AMYLOID ACCUMULATION TRAJECTORIES IN A RISK-ENRICHED ALZHEIMER'S DISEASE COHORT

Recent studies in the field have shown that the early amyloid accumulation pattern may be a critical indicator for improving prediction of cognitive decline (Bilgel et al., 2016; Koscik et al., 2019a). However, in a typical longitudinal neuroimaging data, only a few subjects have brain imaging scans early in their lives, so a model which enables a robust sequential estimation of the early amyloid accumulation pattern is desirable to test various hypotheses based on the estimated measures. Interestingly, we note that our model from the previous chapter could also serve as a way to estimate the early amyloid accumulation trajectory in the unobserved past (e.g., $t < 1$) by "reversing" the direction of the prediction. In this chapter, we use our CRow model from the previous chapter to make sequential predictions of the amyloid measures of an AD cohort and perform a unique longitudinal neuroimaging analysis. The results, while preliminary, are promising and suggest that the measures derived from such a model may enable a better understanding of the early pathological process of AD.

7.1 Overview

We continue our effort from the previous two chapters to better understand the amyloid-beta pathology development which is a defining feature of Alzheimer's disease (AD) (Sperling et al., 2011) and is a primary pathological event leading to cognitive decline and dementia (Hardy and Higgins, 1992; Jack et al., 2018). Specifically, we are interested in cognitively normal *preclinical* individuals who are at a higher risk of developing AD-related dementia with increased cortical amyloid burden measured in vivo with Pittsburgh Compound B (PiB) (Reiman et al., 2009; Pike et al., 2007) showing a greater cognitive decline over time. Thus, in addition to what we

have demonstrated in the previous chapters (Chapter 5 and Chapter 6), there is a great deal of effort in the field to better characterize the longitudinal pattern of amyloid accumulation which will be crucial for effective early detection and intervention of AD (Vlassenko et al., 2011; Sojkova et al., 2011).

A longitudinal study of amyloid accumulation allows us to derive a unique measurement called the *time of onset* (TO), the age at which amyloid accumulates above a critical threshold from PiB- to PiB+. This has been pointed out as a strong indicator of the amyloid burden effect and as one of the earliest signs of AD progression (Koscik et al., 2019a; Brookmeyer et al., 1998). Also, the associations between TO and a well-known genetic risk factor of AD, apolipoprotein E (APOE) ϵ4 allele (Naj et al., 2014), have been investigated (Thambisetty et al., 2013; Naj et al., 2014; Corder et al., 1993). In particular, those with at least one APOE ϵ4 allele (APOE+) showed earlier TOs compared to those with no ϵ4 allele (APOE-) (Khachaturian et al., 2004; Fleisher et al., 2013; Jack et al., 2015).

However, an extensive longitudinal analysis of amyloid accumulation in terms of TO is often limited in practice. One of the main difficulties is that the number of subjects with observed TOs is relatively few since (1) fewer subjects become PiB+ and (2) their scans do not always capture the point of inflection (i.e., PiB- $\rightarrow$ PiB+) if their first scans are acquired after the TO. As a result, several earlier literature have resorted to cross-sectional studies (Fleisher et al., 2013; Naj et al., 2014; Corder et al., 1993; Thambisetty et al., 2013). Recently, studies have found ways to alleviate this by estimating the amyloid accumulation trajectory based on the observed longitudinal trends that were shown via group-based trajectory modeling (Koscik et al., 2019a) and individual-level linear estimation (Bilgel et al., 2016). However, a model which estimates the (1) region-wise amyloid trajectories (2) at the individual-level (3) with the nonlinear trend has not been developed to explicitly consider the variability of the amyloid

Time Points	T = 1	T = 2	T = 3	T = 4	T = All
Number of subjects	63	57	106	8	234
Sex (M/F)	16 / 47	19 / 38	37 / 69	1 / 7	73 / 161
Age (mean/s.d.)	63.0 / 7.0	63.5 / 7.2	63.5 / 6.4	69.4 / 5.3	63.8 / 6.7
Interval years (mean/s.d.)	- / -	3.87 / 2.20	3.42 / 1.37	2.33 / 0.58	3.42 / 1.57
APOE (+/-)	27 / 36	24 / 33	41 / 65	5 / 3	97 / 137

Table 7.1: Demographics of Wisconsin Registry for Alzheimer's Prevention dataset for this study.

accumulation in terms of regions *and* subjects.

Contributions

In this work, we investigate the region-wise amyloid accumulation time of onset (TO) of preclinical AD at the individual-level from a longitudinal PiB cohort. Methodologically, we use a sequential deep neural network model (Hwang et al., 2019c) from the previous chapter to estimate the non-linear amyloid accumulation patterns of each subject in multiple cortical regions measured from longitudinal PiB Positron Emission Tomography (PET) scans. This allows us to estimate the TO for each region *and* subject where we analyze its region-wise patterns and their associations to APOE genotype.

7.2 Methods

Participants

The participant data were acquired from the Wisconsin Registry for Alzheimer's Prevention (WRAP), a cohort of middle-aged adults who are followed longitudinally for [C11] PiB-PET scans we previously analyzed in Chapter 5. From this study, we included the subset of participants who were cognitively unimpaired at the time of scans. On average, the scans were separated by 3.42 years (standard deviation (s.d.) 1.57) with the minimum and maximum interval years of 1.55 and 8.43 respectively. The ages at the

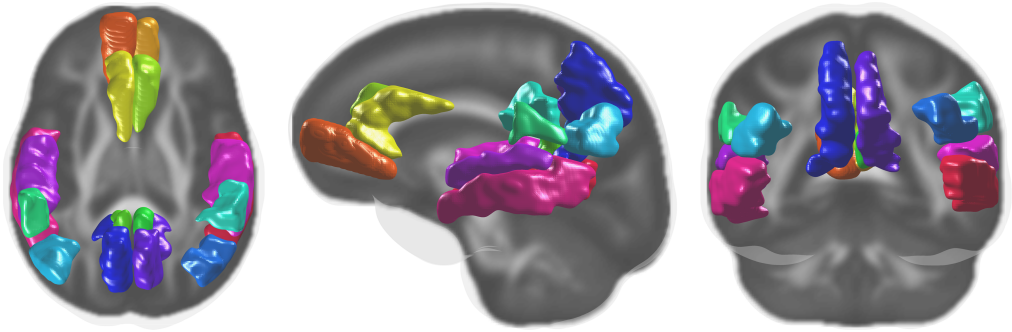


Figure 7.1: Overlays of 16 PiB DVR ROIs.

first scan were of mean 65.88 (s.d. 6.79) years, and 41% of the subjects had at least one ϵ4 allele (APOE+). Letting N_T be the number of subjects with exactly T time points, there were $N_1 = 63$, $N_2 = 57$, $N_3 = 106$, and $N_4 = 8$ subject. The entire cohort of $N_{All} = 234$ subjects consisted of 73 males and 161 females with a mean age of 63.8 (s.d. 6.7). The full demographics are shown in Table 7.1.

PET Imaging and Processing

[C-11] PiB PET scans acquired from the participants were used to reconstruct the PET data using a filtered back-projection algorithm (DIFT) with random event correction, attenuation of annihilation radiation, deadtime, scanner normalization and scatter radiation. Then, they were realigned and coregistered in SPM8 and transformed into voxel-wise distribution volume ratio (DVR) maps using the time activity curve of the cerebellum GM as the reference region. Using SPM8, the DVR images were also spatially normalized to the Montreal Neurological Institute (MNI) space and smoothed with an 8 mm full width at half max Gaussian filter. Further details on the processing are found in (Johnson et al., 2014a). For each time point and subject, we measured the PiB distribution volume ratio (PiB DVR) in 8 bilateral AAL regions and the age at the scan. To derive the region-wise PiB DVR measures, 16 (eight bilateral) of the 116 Automated

Anatomical Labeling (AAL) atlas regions (Tzourio-Mazoyer et al., 2002) that are implicated as important in AD (Clark et al., 2016) based on the global amyloid burden (Sprecher et al., 2015) were used and are shown in Fig 7.1. For each bilateral AAL region pair (left and right hemispheres), the combined PiB-DVR was measured by taking the average of the regions weighted by their corresponding volumes. The spaghetti plots of the ROIs are shown in Fig. 7.2 where red lines are APOE+ subjects and blue lines are APOE- subjects. For each ROI, the PiB+ thresholds are computed individually (shown as cyan lines in Fig. 7.2) which we describe later.

Sequential Deep Neural Network for PiB-DVR Trajectory Estimation

We estimated the PiB DVR of the subjects before their observed scans by *retrospectively* using our sequential deep neural network model, Conditional Recurrent Flow (CRow), described in Chapter 6.

Conditional Recurrent Flow

Briefly, given a sequence of input/output pairs $\{\mathbf{u}^t, \mathbf{v}^t\}$ for $t = 1, \ldots, T$ time points, modeling the relationship between the variables across time needs to also account for the temporal characteristic of the sequence. CRow is an *invertible network which is also recurrent* such that given a *sequence* of inputs $\{\mathbf{u}^t\}$ (i.e., features $\{\mathbf{x}^t\}$) and their *sequence* of outputs $\{\mathbf{v}^t\}$ (i.e., covariates/labels and latent information $\{\mathbf{y}^t, \mathbf{z}^t\}$), we can model the invertible relationship between those sequences for posterior estimation and conditional sequence generation as illustrated in Fig. 1.8. As we saw in Chapter 6, CRow is a *sequential generative model* which can generate sequential samples given a sequence of conditions. In this sense, we use CRow by setting the conditions to be the ages and the sequential samples that

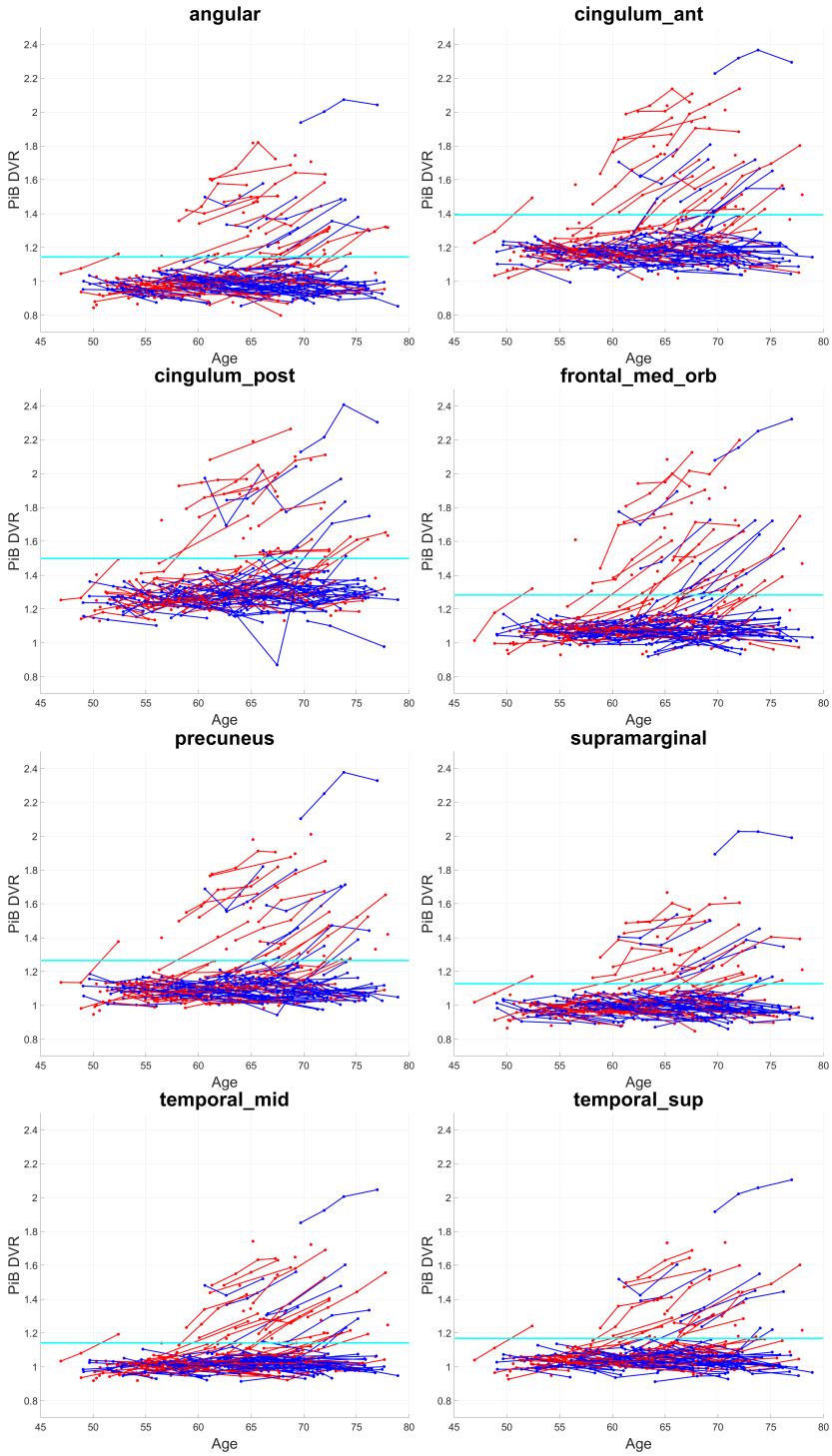


Figure 7.2: Observed PiB DVR Trajectories of 8 combined bilateral ROIs. Red: APOE+, Blue: APOE-, Cyan: ROI-specific thresholds as shown in Table 7.2.

we generate to be the PiB DVR measures that we want to estimate that correspond to the given ages.

PiB Trajectory Estimation

Now, we see how the model can be set up to estimate PiB trajectories. Each sample is a sequence of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_T$ of a subject scanned at ages $\mathbf{y}_1, \ldots, \mathbf{y}_T$ where each $\mathbf{x}_t \in \mathbb{R}^8$ is an 8 combined bilateral PiB DVRs at the time point t. Thus, once the model is trained, it is able to forward and inverse map from the PiB DVR sequences and their corresponding ages at scans:

$$\mathbf{x}_1, \ldots, \mathbf{x}_T \leftrightarrow \mathbf{y}_1, \ldots, \mathbf{y}_T. \tag{7.1}$$

Since the model is recursive, we can use all longitudinal samples of varying lengths. Note that all the sequences have the baseline scan at $t = 1$ (i.e., the earliest scan) and the most recent scan at $t = T$ (i.e., $\mathbf{y}_1 < \cdots < \mathbf{y}_T$).

Now, the goal is to estimate the PiB DVRs of a subject retrospectively *before* the baseline scan (e.g., $\mathbf{x}_t$ for $t < 1$):

$$\mathbf{y}_{T'}, \ldots, \mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_T \rightarrow \mathbf{x}_{T'}, \ldots, \mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_T \tag{7.2}$$

where $\mathbf{y}_{T'}$ is the earliest (youngest) age that we use to estimate the corresponding PiB DVR ($\mathbf{x}_{T'}$). For instance, given a sample with scans $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ at $\mathbf{y}_1 = 60, \mathbf{y}_2 = 65, \mathbf{y}_3 = 70$, we estimate the PiB DVRs retrospectively until age 45:

$$\mathbf{y}_{-2} = 45, \mathbf{y}_{-1} = 50, \mathbf{y}_0 = 55 \rightarrow \mathbf{x}_{-2}, \mathbf{x}_{-1}, \mathbf{x}_0.$$

Specifically, we recursively estimate the retrospective PiB DVRs of all the subjects from their baseline scans until age 45 ($\mathbf{y}_{T'} = 45$) with a scan interval of 2.5 years (i.e., estimate every 2.5 years). Thus, depending on the age of the baseline scan, subjects may have a varying number of estimated PiB DVR time points.

ROI Name	PiB DVR Threshold
angular	1.144
cingulum_ant	1.393
cingulum_post	1.498
frontal_med_orb	1.283
precuneus	1.265
supramarginal	1.128
temporal_mid	1.142
temporal_sup	1.169

Table 7.2: ROI-specific PiB DVR thresholds. We individually derived the PiB DVR thresholds where the PiB DVR above the threshold is considered to be PiB+ associated with a higher risk of cognitive decline. Visualized in Fig. 7.2.

Time of Onset (TO) Computation

Now, once we estimate the PiB DVRs of all the subjects up to age 45 in 8 combined ROIs, we compute the time of onset (TO) as follows. First, based on a previous work which identified the PiB burden value of 1.2 to be the cut-off that maximized both sensitivity and specificity of receiver operating characteristic analysis of the expert visual ratings of PiB+ or PiB- (Racine et al., 2016), a subject is considered to be PiB+ when the subject's global PiB DVR is above 1.2. If the global PiB DVR is below 1.2, the subject is considered PiB-. Then, from the entire cohort, we select those who remain PiB- throughout all their scans. In this longitudinal PiB- group, we estimated that the global threshold of 1.2 is very close to the mean global PiB DVR plus 3 times the standard deviation of the global PiB DVRs. Thus, we applied the same formula to each ROI such that for each of the 8 (combined) PiB ROIs, we set the *ROI-specific threshold* as the mean plus 3 times the standard deviation of the PiB DVRs of that specific ROI. This was a reasonable approach for ROI-based analyses since the ROIs had varying distributions of PiB DVRs. The ROI-specific threshold values are shown in Table 7.2.

Using these ROI-specific thresholds, we computed the TOs which either the estimated or the observed PiB DVRs crossed the thresholds (i.e., the

ROI became PiB+). Specifically, for each ROI, we found a pair $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$ which the PiB DVR accumulated from $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$ (corresponding age from $\mathbf{y}_{t-1}$ to $\mathbf{y}_t$). Then, although the model could technically predict $\mathbf{x}_t$ for any given age $\mathbf{y}_t$, for the purpose of this preliminary analysis, we find it reasonable to assume a linear accumulation pattern between a short gap of predictions (i.e., 2.5 years). We note that if the accumulation pattern appears highly nonlinear between the predictions, we could make even finer grained predictions (e.g., 0.1 years gap). Then, we computed the time of onset $\mathbf{y}_{onset}$ crossing the threshold as follows:

$$\mathbf{y}_{onset} = \frac{\text{threshold} - \mathbf{x}_{t-1}}{\mathbf{x}_t - \mathbf{x}_{t-1}}(\mathbf{y}_t - \mathbf{y}_{t-1}) + \mathbf{y}_{t-1} \tag{7.3}$$

which essentially finds the point of inflection ($\mathbf{y}_{onset}$) when the PiB DVR crosses the corresponding threshold. For those who were still PiB+ at age 45, we bounded their time of onset to be 45.

For those subjects that never became PiB+ (i.e., PiB- throughout the sequence), we used the Wisconsin Life Expectancy Table (`www.dhs.wisconsin.gov/stats/life-expectancy.htm`) to speculate the time of onset to be the expected life expectancy. Specifically, the TO would be the expected life range which is $\mathbf{y}_T$ (the latest scan age) plus the life expectancy corresponding to the age group and gender. A survival analysis method such as the life expectancy estimation is a reasonable option found in other studies since it is known that a consistent percentage of the subjects do eventually become PiB+ (Bilgel et al., 2016; Koscik et al., 2019b).

Statistical Analysis

We looked at the association between the amyloid accumulation pattern measured in TOs and a well-known genetic risk factor apolipoprotein E (APOE) (Naj et al., 2014). Specifically, it is known that individuals carrying the $\epsilon 4$ allele are at increased risk of AD compared with those carrying no $\epsilon 4$ alleles. Thus, we specify two groups based on the APOE

ROI	Mean TO (APOE+ / APOE-)	p-value
angular	76.5 / 81.7	*0.0001
cingulum_ant	76.4 / 82.2	*0.0001
cingulum_post	77.4 / 82.0	*0.0004
frontal_med_orb	76.1 / 81.6	*0.0001
precuneus	76.9 / 81.6	*0.0003
supramarginal	77.4 / 82.0	*0.0004
temporal_mid	77.4 / 81.1	*0.0061
temporal_sup	77.6 / 81.8	*0.0012

Table 7.3: Mean time of onset and the group difference results in each ROI. $*$ indicates statistical significance after the Bonferroni correction.

ϵ allele status: the APOE+ (ϵ4-allele) and APOE- (no ϵ4-allele) groups. Then, for each region, we tested the difference of TOs between the APOE+ and APOE- groups using the two-sample t-test. We used the Bonferroni corrected significance threshold ($\alpha = 0.05$) for the type-1 error correction.

7.3 Results

The PiB trajectory estimation results in Fig. 7.3 show the predicted trajectories (dashed lines) given the original trajectories (solid lines). Red and blue lines are APOE+ and APOE- subjects respectively. For each ROI, the TOs were computed based on the ROI-specific (Table 7.2), so each subject had one TO for each ROI. The average time of onset of the APOE+ and APOE- groups for each ROI are shown in Table 7.3. The statistical significance of the group differences between APOE+ and APOE- are also shown in Table 7.3 with the Bonferroni correction. We found that for all 8 ROIs, the differences of time of onset between APOE+ and APOE- were statistically significant (p-value < 0.05) with the lower mean time of onset of APOE+ subjects. Overall, the estimated trajectories show linear patterns after the PiB DVR reaches a critical point. This is more apparent in Fig. 7.4 which shows the subjects with $\mathsf{T} = 4$ time points which inform the longitudinal patterns the most with long observed sequences.

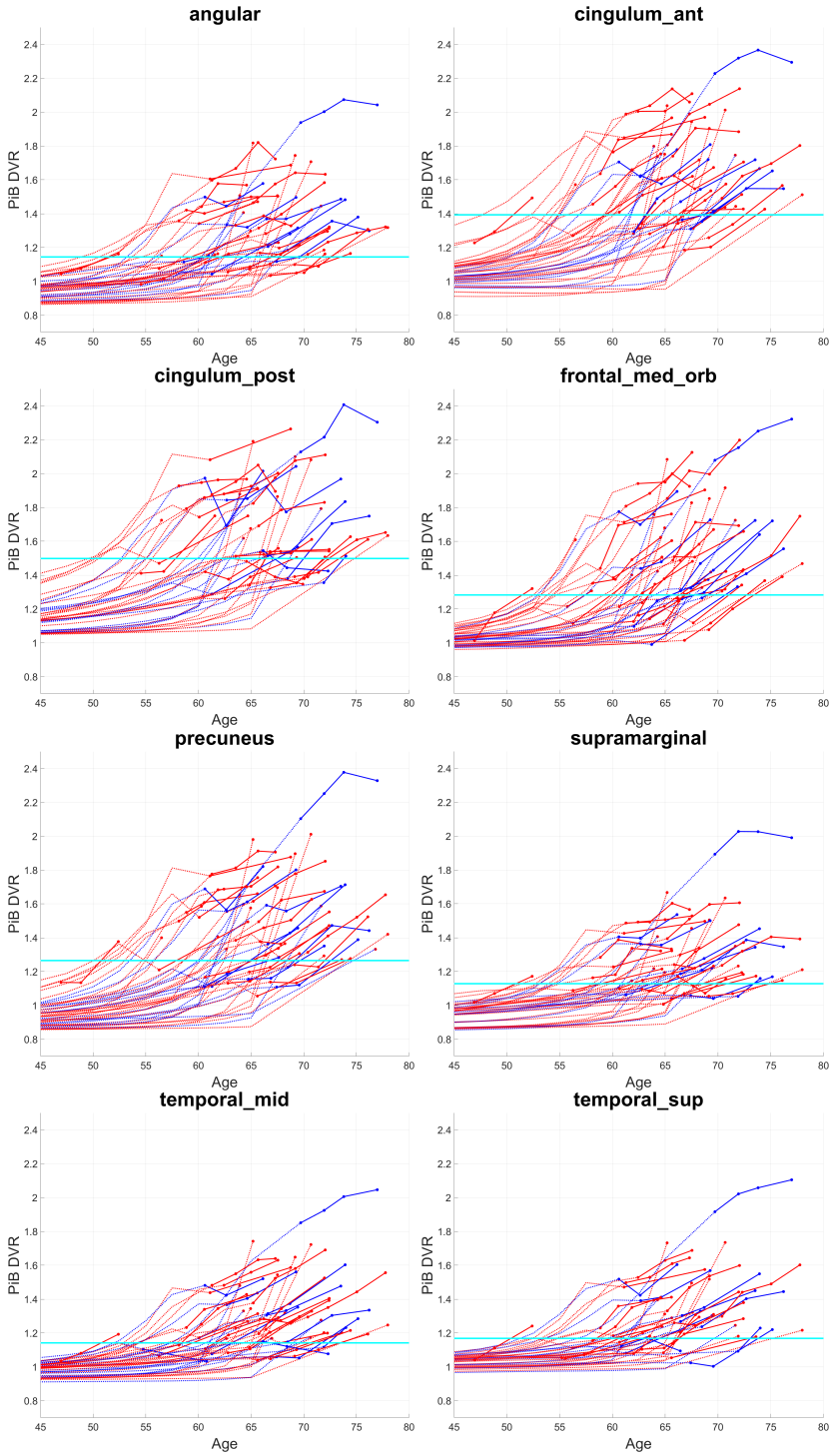


Figure 7.3: Retrospective PiB DVR trajectory estimation. Straight lines: observed PiB DVR trajectories. Dashed lines: Estimated PiB DVR trajectories. Right two columns: The observed PiB trajectories are shifted with respect to their time of onset. Only showing the PiB+ subjects.

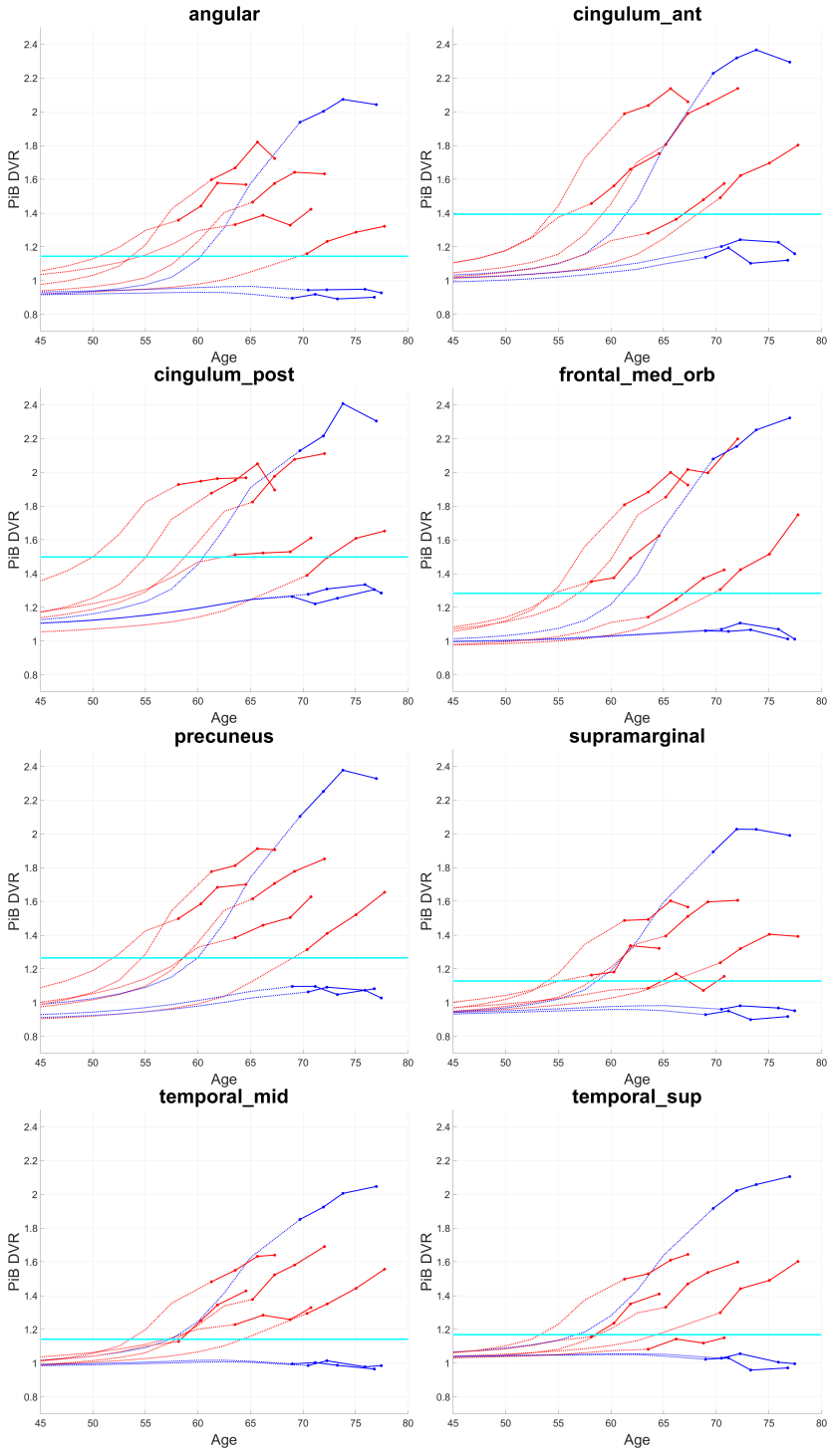


Figure 7.4: Retrospective PiB DVR trajectory estimation of T = 4 subjects.

We also plot the PiB DVR trajectories from a different perspective just like how (Koscik et al., 2019a) presented it in their work. Instead of using the given ages as the independent variable (x-axis), we used the TOs to "shift" the observed PiB DVRs (solid lines) as shown in Fig. 7.5. This allows us to observe the subjects' amyloid accumulation patterns *after* they become PiB+. Thus, if there is a consistent pattern of amyloid accumulation, it may imply that the amyloid accumulation pattern is *independent* of age but *dependent* of their time of onset which is consistent across the subjects. In particular, we are interested in seeing whether such "cohort-level" amyloid accumulation trend follows that of the widely accepted amyloid-tau-neurodegeneration (ATN) curve hypothesizing the abnormality progressions of various biomarkers including amyloid beta (Jack et al., 2016). In these new plots, the x-axes are the "Years since PiB+" where 0 is the point of inflection (i.e., time of onset). Thus, the PiB- subjects will have negative "Years since PiB+" values.

We next narrow the scope of the analysis that for each ROI, we only observe the samples that became PiB+ (i.e., excluding the PiB- samples that never become PiB+). This is a smaller subset for each ROI that focuses on the PiB+ group only. Note that the ROIs can have a varying number of samples since only some of the ROIs may become PiB+ for the same subject. Table 7.4 shows the mean TOs for the APOE+ and APOE- groups where we see much smaller differences. No ROIs had significant differences when the TOs of the APOE+ and APOE- groups were compared.

In Fig. 7.6, we see the box plots of the TOs of the APOE+ and APOE- samples from the PiB+ group. We then show the box plots of the APOE+ (Fig. 7.7) and APOE- (Fig. 7.8) groups *separately*.

We also looked into the order at which these ROIs become PiB+. In Table 7.5, we show the distribution of each ROI in terms of its order when it became PiB+. Each column indicates the order at which the corresponding ROI became PiB+ with respect to the other ROIs. Specifically, for a subject,

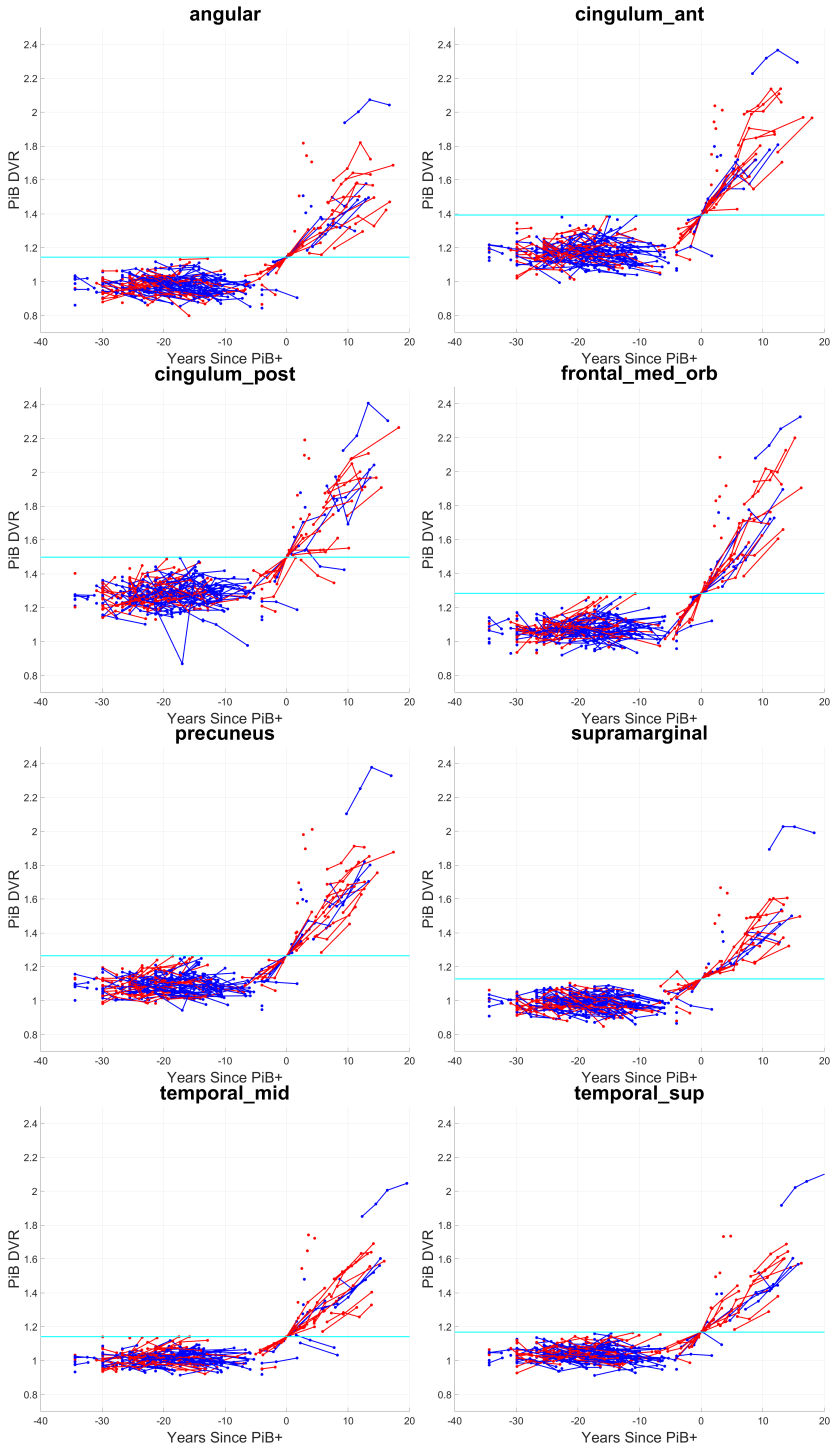


Figure 7.5: Years since PiB+ vs. PiB DVR for each ROI. The ages show in Fig. 7.3 are shifted by the estimated TOs. Thus, the point of inflection at which the ROI becomes PiB+ is at 0 on the x-axis. Only the observed PiB DVRs are shown.

suppose we have the TO of each ROI, i.e., TO_k is the TO of the ROI k. Then, we order the ROIs based on their TOs from the smallest TO_k to the highest TO_k. We say that the ROI k has the ith onset if its TO is the ith smallest one in the list, i.e., TO_k is the ith smallest. So for each ROI (row), the number for each column indicates the probability of that ROI having the ith onset within our data. For instance, temporal_mid has the 1st onset for 25% of the time while it has the 5th onset for only 3% of the time. Thus, if an ROI has a high % in an ith column, then it implies the ROI consistently had the ith onset. Table 7.6 shows the results for the APOE+ and APOE- groups separately.

7.4 Discussion

In this work, we looked for the association between the time of onset (TO) of amyloid pathology across 8 AD-related regions and APOE genotype. For those subjects without an observable TO when they become PiB+, we used a sequential deep neural network model from Chapter 6 to capture the trajectory pattern with a temporal latent representation to (1) retrospectively estimate their ROI-wise PiB DVRs ($\mathbf{x}_{t<1} \in \mathbb{R}^8$ for 8 ROIs) before the observed first scan $\mathbf{x}_1$ and (2) compute the TO for each ROI based on the ROI-specific thresholds. As shown in Table 7.3, the TOs between the APOE+ and APOE- groups were different where the APOE+ group had earlier average TOs across all the ROIs (76.1 to 77.6) while the APOE- group had their average time of onset approximately 4 to 5 years later than those of APOE+ (81.1 to 82.2). This implied that the APOE+ group showed a strong association with an earlier TO of amyloid accumulation across the ROIs. We also observed some variability of TOs among the ROIs. For instance, in frontal_med_orb, the APOE+ group has an average time of onset of 76.1 while in temporal_sup, the average TO is 77.6, showing over a year of difference between the TOs of these ROIs.

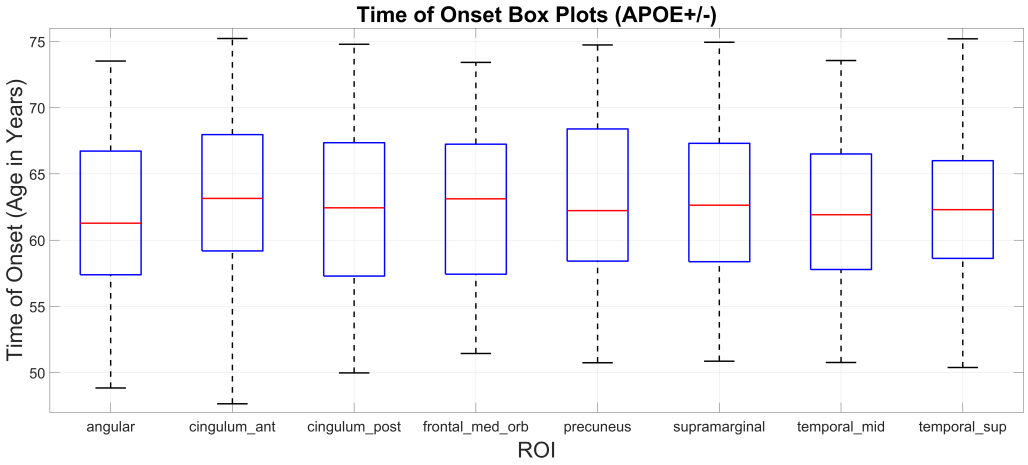


Figure 7.6: Box plot of TOs of APOE+ and APOE-. For each ROI, the top and bottom bars are the max and min TOs of that ROI respectively. The box indicates the standard deviation from the mean which is the red line.

For cross-sectional settings, the association between APOE $\epsilon 4$ allele status and amyloid accumulation has been shown in previous studies (Fleisher et al., 2013; Naj et al., 2014; Corder et al., 1993; Thambisetty et al., 2013). These cross-sectional studies have also observed that the APOE+ group showed early amyloid positivity compared to the APOE- group. A recent longitudinal study (Bilgel et al., 2016) had estimated the linear patterns of the amyloid accumulation at the individual-level which also showed the association between the TOs of the individuals to the APOE4 status. Their analysis estimated the TO to be age 64 for the APOE+ group and age 76 for the APOE- group showing nearly a 12 year difference. Note that the average baseline scan age of their cohort was 77.1 (7.8 s.d.) while the baseline scan age of our cohort was 65.9 (6.8 s.d.). Our cohort was nearly 11 years younger than that of (Bilgel et al., 2016) in terms of the average baseline scan ages. Thus, the proportion of the late onset subjects with high amyloid later in life in our cohort was smaller than that of (Bilgel et al., 2016). Thus, in our case, many of the subjects whose TOs were not observed had their expected life years as surrogates, resulting in relatively high average TOs for both the APOE+ and APOE- groups. Still,

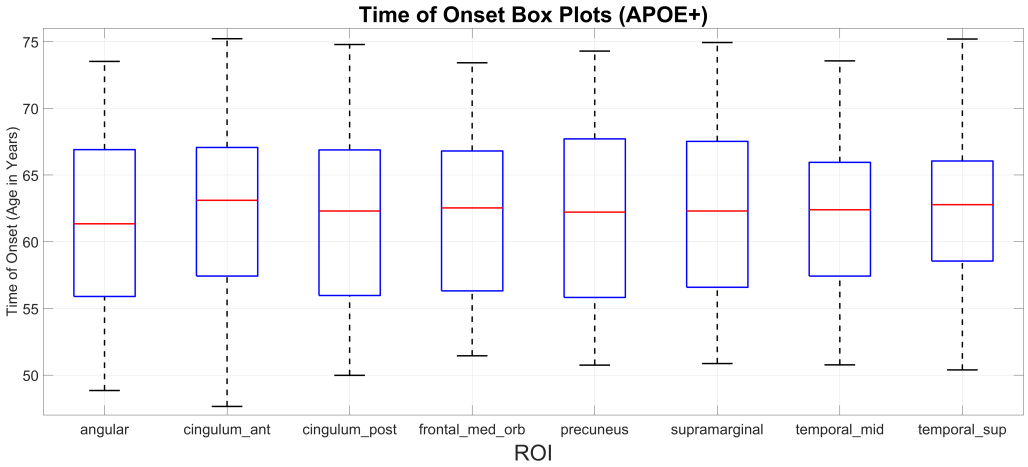


Figure 7.7: Box plot of TOs of APOE+.

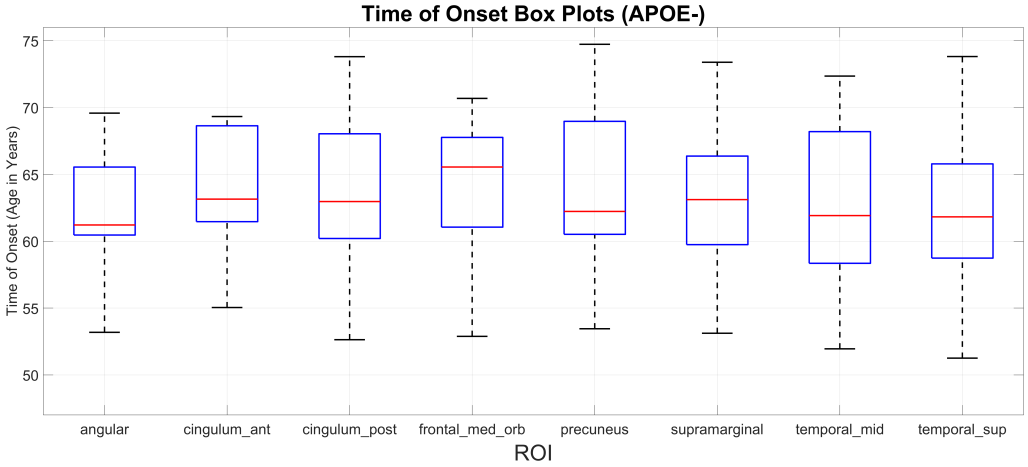


Figure 7.8: Box plot of TOs of APOE-.

the estimated TOs contributed strongly enough to result in a statistically significant difference between the TOs between the APOE+ and APOE- groups.

Another notable study was by (Koscik et al., 2019a) which involved the same WRAP study (with a slightly different cohort due to different versions of the WRAP database). Their group-based trajectory modeling showed the time of onset of ~50.6 for the "early onset" group, ~61.6 for the "intermediate onset" group, and ~71.3 for the "late onset" group. Unlike

ROI	Mean TO (APOE+ / APOE-)
angular	61.6 / 62.2
cingulum_ant	62.2 / 63.7
cingulum_post	62.3 / 63.4
frontal_med_orb	62.2 / 64.1
precuneus	62.2 / 64.2
supramarginal	62.4 / 63.2
temporal_mid	62.2 / 62.6
temporal_sup	62.0 / 62.5

Table 7.4: Mean TO of APOE+ and APOE- Groups with PiB+ Cohort Only. For each ROI, this is conditioned on those who eventually became PiB+.

our analysis, the PiB- subjects were labeled as "Non-accumulators", and the statistical analyses used the linear combinations of the TOs of these onset groups to look for the individual-level TOs and their associations to other risk factors. They also found strong associations between the APOE $\epsilon 4$ carriers and the time of onset.

Similar to these two previous studies, our ROI-based analyses showed consistent results regarding the relations of TO to the APOE $\epsilon 4$ allele status. Even though the ROI-specific thresholds were computed in a data-driven manner, the consistent average times of onset across the ROIs indicate that the amyloid accumulation patterns may be similar among the ROIs. Since the PiB DVR measures were not globally averaged, one of the difficulties of the ROI-based analysis was that the individual measurements of the ROIs were in general noisier than the global PiB DVR. Still, we were able to stably map the PiB DVR patterns to age via the bidirectional training mechanism of our model (see Chapter 6) and make robust predictions.

One of the advantages of our ROI-based trajectory estimation is how we can observe the patterns with respect to the individual ROIs. In Fig. 7.7 we observe that the mean TOs across the ROIs of APOE+ subjects are less variable and distinct compared to the TOs of APOE- subjects shown in Fig. 7.8 which have smaller deviations as well. This may suggest that the amyloid accumulation patterns of APOE- subjects are more consistent

ROI	% of ROIs becoming PiB+ at ith Order							
	i=1	2	3	4	5	6	7	8
angular	0.14	0.12	**0.16**	**0.16**	**0.16**	0.14	0.09	0.02
cingulum_ant	0.11	0.13	0.09	0.11	**0.17**	0.13	0.11	**0.17**
cingulum_post	0.10	0.06	0.08	0.18	0.06	**0.22**	0.10	0.20
frontal_med_orb	0.13	**0.17**	0.04	0.09	**0.17**	0.13	**0.17**	0.11
precuneus	0.03	0.11	0.16	0.18	**0.21**	0.11	0.16	0.05
supramarginal	0.07	0.05	**0.23**	0.16	0.14	0.07	**0.23**	0.07
temporal_mid	**0.25**	0.23	0.10	0.10	0.03	0.08	0.10	0.13
temporal_sup	**0.24**	0.18	0.06	0.09	0.12	0.03	0.09	0.18

Table 7.5: Distribution of ROIs becoming PiB+ at ith Order for APOE+ and APOE-. Each row shows the probability that the ROI is the ith ROI to become PiB+. For instance, temporal_sup became PiB+ first ($i = 1$) in 24% of the cases. For each ROI, only the subjects who eventually became PiB+ are included. For each row (ROI), the probabilities sum up to 1 since it is a distribution with respect to the corresponding ROI.

across the ROIs and subjects while the APOE+ subjects have more irregular pattern of spread across the ROIs.

In Table 7.5 showing the order of ROI onsets of the APOE+ and APOE- groups, we see that temporal_mid and temporal_sup become the first PiB+ ROIs for nearly 25% of the time. Other than those 2 ROIs, we do not particularly see an ROI predominantly becoming PiB+ at the ith order. In Table 7.6, we show the APOE+ group at the top and the APOE- group at the bottom. For the APOE- group (Bottom), the ROIs show clearer patterns with the ROIs becoming PiB+ more dominantly. Specifically, we see that higher %'s are seen in the rows which imply that the ROIs become PiB+ more consistently at the ith order. For instance, precuneus was the fifth ROI to became PiB+ (i.e., i=5) for 42% of the time which is the most consistent ordering of PiB+ throughout the ROIs. On the other hand, for the APOE+ group (Top), the ROIs become PiB+ with less consistent patterns as each ROI does not have a dominant ordering (i.e., high % in a specific i). This may further imply that for the APOE+ subjects, their

ROI	% of ROIs becoming PiB+ at ith Order (APOE+)							
	i=1	2	3	4	5	6	7	8
angular	0.16	0.13	0.09	**0.19**	0.16	0.13	0.13	0.03
cingulum_ant	0.14	0.14	0.11	0.14	**0.20**	0.11	0.00	0.14
cingulum_post	0.11	0.09	0.09	0.14	0.06	0.17	0.14	**0.20**
frontal_med_orb	0.13	**0.19**	0.06	0.10	0.13	**0.19**	0.13	0.06
precuneus	0.04	0.12	**0.23**	0.15	0.12	0.12	0.19	0.04
supramarginal	0.04	0.07	**0.18**	**0.25**	0.18	0.04	0.18	0.07
temporal_mid	**0.22**	**0.22**	0.11	0.04	0.04	0.11	0.11	0.15
temporal_sup	**0.23**	0.14	0.05	0.09	0.14	0.05	0.09	**0.23**

ROI	% of ROIs becoming PiB+ at ith Order (APOE-)							
	i=1	2	3	4	5	6	7	8
angular	0.09	0.09	**0.36**	0.09	0.18	0.18	0.00	0.00
cingulum_ant	0.00	0.08	0.00	0.00	0.08	0.17	**0.42**	0.25
cingulum_post	0.07	0.00	0.07	0.27	0.07	**0.33**	0.00	0.20
frontal_med_orb	0.13	0.13	0.00	0.06	**0.25**	0.00	**0.25**	0.19
precuneus	0.00	0.08	0.00	0.25	**0.42**	0.08	0.08	0.08
supramarginal	0.13	0.00	**0.31**	0.00	0.06	0.13	**0.31**	0.06
temporal_mid	**0.31**	0.23	0.08	0.23	0.00	0.00	0.08	0.08
temporal_sup	**0.27**	**0.27**	0.09	0.09	0.09	0.00	0.09	0.09

Table 7.6: Distribution of ROIs becoming PiB+ at ith Order for APOE+ (Top) and APOE- (Bottom). For each ROI, only the subjects who eventually became PiB+ are included.

amyloid accumulations are occurring across the ROIs more irregularly.

There are several limitations to our analysis. First, we have relatively fewer late onset samples which largely consist of APOE- subjects, so estimating the PiB trajectory patterns of late onset samples is still limited. Compared to the cohort in the study by (Bilgel et al., 2016) which has a large group of late onset subjects, our analysis is focused on mid-to-late onset subjects who are largely APOE+. Thus, the addition of late onset cohort which often consists more of APOE- subjects may provide a better

conditional analysis (i.e., PiB+ only) in the future. Second, related to the previous limitation, our analysis is largely influenced by the PiB- subjects and their estimated TOs based on the life expectancy table which may be an oversimplification. Third, it is still difficult to accurately estimate the trajectories of individuals with only one time point. Based on our model, their predictions are often too "steep" due to the lack of observed longitudinal information to inform the initial trend. This could be alleviated if we incorporate additional covariates other than age which may provide auxiliary longitudinal information.

7.5 Summary

In this chapter, we analyzed the effect of TO of each ROI to APOE genotype at the individual-level. Since the existing cohort often does not have scans that directly reveal TO, we used our sequential invertible neural network from Chapter 6 derive the temporal latent representations and retrospectively estimate the ROI-wise and subject-wise PiB DVRs, and then we computed ROI-specific TOs for the subsequent group analyses. We found significant differences in TOs between the APOE+ and APOE- groups across all 8 ROIs, but we did not find significant differences when we only considered the PiB+ subjects. There are several immediate future works including the analyses on other datasets such as ADNI and performing a correlation analysis with the TOs of (Koscik et al., 2019a) which studied the same cohort. This chapter demonstrated how the CRow formulation could also be applied to an important neuroimaging problem involving sequential brain imaging measures and to understand the early pathological process of Alzheimer's disease.

8 CONCLUSIONS

In this thesis, we showed how we can develop statistical and machine learning models that can learn latent representations and understand various relationships as shown in Fig. 1.4 while effectively addressing data- and domain-specific challenges. From computer vision to neuroimaging, the problems that we tackled involved various types of data modalities ranging from cross-sectional to temporal imaging data and both natural and brain images of different types. Through understanding and modeling relationships of various types with latent representations, we demonstrated how solving those problems may lead to much more impactful outcomes beyond what was possible with the original data directly.

8.1 Contributions

In each chapter, we addressed diverse data- and domain-specific challenges in the following various respects: (1) structure of data, (2) relationship type, (3) problem type, and (4) domain-specific challenges. Specifically,

1. We developed a multi-relational tensor factorization approach to derive the latent representations of objects and predicates to understand the visual relationships between objects in images (Hwang et al., 2018). Our robust tensor formulation robustly captured the sparsely observed visual relationships as a shallow model on its own and also regularized a deep model to achieve state-of-the-art performance on various visual relationship detection tasks.

2. We characterized the longitudinal and cross-sectional progression of brain networks of preclinical Alzheimer's disease (AD) subjects by deriving the coupled harmonic basis of the brain networks (Hwang et al., 2016). Due to the subtle progressive trends of asymptomatic AD

subjects, we imposed the within-subject progression (longitudinal) and across-subject progression (cross-sectional) in the latent representation space (harmonic basis) directly.

3. Combining the classical statistical properties of exponential families and powerful recurrent neural network variants, we proposed a sequential neural network called Sampling-free Probabilistic Gated Recurrent Unit (SP-GRU) which deterministically estimated the uncertainty of all the weights and neurons of GRU (Hwang et al., 2019b). This allowed fast uncertainty quantification of sequential predictions as a byproduct of GRU without costly sampling procedures for high-dimensional sequence prediction tasks and the normative modeling of preclinical Alzheimer's disease cohort for outlier detection.

4. We constructed a generative sequential neural network called CRow (Hwang et al., 2019c) for conditionally generating sequential samples. Building upon an invertible neural network, the model incorporated recurrent subnetworks and temporal context gating to learn the conditional distribution of training data in a latent space and generated sequential samples given observable conditions. In neuroimaging applications which involve small sample sizes, we showed how the realistically generated sequential brain imaging measures could result in statistical analyses consistent with those reported by other studies in the aging literature.

5. We demonstrated how our CRow formulation from the previous chapter could be used to retrospectively estimate the progression of AD pathology. Specifically, we predicted the amyloid accumulation patterns from the PiB PET scans and estimated the time of onset, the age at which the amyloid load surpasses a certain critical threshold. Our model allowed an individual-level and region-level prediction in a nonlinear manner in which previous studies have not attempted.

The code repository can be found in `https://github.com/shwang54`.

8.2 Future Directions

It is clear that a wide range of problems in computer vision and neuroimaging which would benefit from appropriately derived latent representations. The development of powerful methods to extract useful knowledge from data that is generated from scientific and biomedical studies will continue to play a critical role in data science, and some of these developments may partially benefit from the work described in this thesis. We discuss a few short to medium future directions of our research and conclude the thesis.

Further Development of Amyloid Chronicity Analysis

The last chapter of the thesis on the amyloid time of onset analysis is still ongoing work, and the results presented come from preliminary experiments. There are several next steps in the short term. First, we plan to test our hypotheses on other longitudinal neuroimaging datasets including the ADNI dataset that contains longitudinal amyloid load measurements (see Chapter 6 for a brief description of ADNI). Since the amyloid accumulations are observed with a different radiotracer (AV45) from a new cohort, investigating how our model performs on this new set of subjects will be an excellent first step to further validate our model's performance. Second, we will compare our estimated time of onset with those estimated by (Koscik et al., 2019a) where the authors analyzed the same WRAP cohort. This will allow us to test the validity of both methods by (Koscik et al., 2019a). If they are highly correlated with each other and find similar associations with the APOE status, this could imply that the same data with different analytical tools consistently lead to similar findings, further strengthening our hypothesis. There are several other improvements we

plan to incorporate in our analysis including an improved derivation of ROI-specific thresholds.

Longitudinal Neurodegenerative Disease Study: "-Omics", Images and More

Understanding neurodegenerative diseases often involves the complex amalgamation of multiple risk factors. Longitudinal studies integrating various "omics" such as genomics and metabolomics, and other risk factors are showing promising new findings. In addition, a huge collaborative effort by Alzheimer's Disease Neuroimaging Initiative (ADNI) is collecting longitudinal samples of multiple AD related data types, e.g., images and biospecimens. Such discoveries encourage the community to continuously explore and put efforts in accumulating invaluable data, in a positive feedback cycle. Going beyond the biomarkers we have used in this thesis so far, our future aims are to expand the analyses to known and new risk factors in multiple modalities. In fact, we are currently investigating a multi-modal brain network model which may allow us to understand a disease pathology propagation pattern by combining both the structural DTI and PiB PET scans. Also, we plan to expand the scope of applications to other neurodegenerative diseases and disorders such as Parkinson's disease and childhood brain disorders.

REFERENCES

Abbasnejad, M Ehsan, Anthony Dick, and Anton van den Hengel. 2017. Infinite variational autoencoder for semi-supervised learning. In *Cvpr*, 781–790. IEEE.

Abdi, Hervé, and Lynne J Williams. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics* 2(4):433–459.

Absil, P-A, Robert Mahony, and Rodolphe Sepulchre. 2009. *Optimization algorithms on matrix manifolds*. Princeton University Press.

Achard, Sophie, Raymond Salvador, Brandon Whitcher, et al. 2006. A resilient, low-frequency, small-world human brain functional network with highly connected association cortical hubs. *The Journal of neuroscience* 26(1):63–72.

Aguiar, A Pedro, and Joao P Hespanha. 2007. Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty. *IEEE transactions on automatic control* 52(8):1362–1379.

Akselrod, Solange, David Gordon, F Andrew Ubel, et al. 1981. Power spectrum analysis of heart rate fluctuation: a quantitative probe of beat-to-beat cardiovascular control. *Science* 213(4504):220–222.

Alberti, Marina, John Folkesson, and Patric Jensfelt. 2014. Relational approaches for joint object classification and scene similarity measurement in indoor environments. In *Aaai 2014 spring symposia: Qualitative representations for robots*.

Alexander, Gene E, Kewei Chen, Pietro Pietrini, et al. 2002. Longitudinal PET evaluation of cerebral metabolic decline in dementia: a potential

outcome measure in Alzheimer's disease treatment studies. *American Journal of Psychiatry* 159(5):738–745.

Amodei, Dario, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *Icml*.

Anandkumar, Animashree, Rong Ge, Daniel Hsu, et al. 2014. Tensor decompositions for learning latent variable models. *JMLR* 15.

Andreas, Jacob, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Deep compositional question answering with neural module networks. In *Cvpr*.

Antol, Stanislaw, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the ieee international conference on computer vision*, 2425–2433.

Ardizzone, Lynton, Jakob Kruse, Sebastian Wirkert, Daniel Rahner, Eric W Pellegrini, Ralf S Klessen, Lena Maier-Hein, Carsten Rother, and Ullrich Köthe. 2018. Analyzing inverse problems with invertible neural networks. *arXiv preprint arXiv:1808.04730*.

Atzmon, Yuval, Jonathan Berant, Vahid Kezami, et al. 2016. Learning to generalize to new compositions in image understanding. *arXiv preprint arXiv:1608.07639*.

Baddeley, AD, S Bressi, Sergio DELLA SALA, Robert LOGIE, and H Spinnler. 1991. The decline of working memory in Alzheimer's disease: A longitudinal study. *Brain* 114(6):2521–2542.

Bansal, Trapit, Chiranjib Bhattacharyya, and Ravindran Kannan. 2014. A provable SVD-based algorithm for learning topics in dominant admixture corpus. In *Nips*.

Bartlett, MS. 1963. The spectral analysis of point processes. *Journal of the Royal Statistical Society: Series B (Methodological)* 25(2):264–281.

Bathe, Klaus-Jürgen, and Edward L Wilson. 1973. Solution methods for eigenvalue problems in structural mechanics. *International Journal for Numerical Methods in Engineering* 6(2):213–226.

Bertsekas, Dimitri P. 1999. *Nonlinear programming*. Belmont (Mass.): Athena Scientific.

Bilgel, Murat, Yang An, Yun Zhou, Dean F Wong, Jerry L Prince, Luigi Ferrucci, and Susan M Resnick. 2016. Individual estimates of age at detectable amyloid onset for risk factor assessment. *Alzheimer's & Dementia* 12(4):373–379.

Bińkowski, Mikołaj, Dougal J Sutherland, Michael Arbel, and Arthur Gretton. 2018. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*.

Blundell, Charles, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*.

Boráros, Bc Peter, and Bc Peter Boráros. 1969. Network anomaly detection by means of spectral analysis. *Technometrics* 11(1):1–21.

Boutsidis, Christos, and Efstratios Gallopoulos. 2008. SVD based initialization: A head start for nonnegative matrix factorization. *Pattern Recognition* 41.

Brookmeyer, Ron, Sarah Gray, and Claudia Kawas. 1998. Projections of alzheimer's disease in the united states and the public health impact of delaying disease onset. *American journal of public health* 88(9):1337–1342.

Cairns, Nigel J, Richard J Perrin, Erin E Franklin, et al. 2015. Neuropathologic assessment of participants in two multi-center longitudinal observational studies: The Alzheimer Disease Neuroimaging Initiative (ADNI) and the Dominantly Inherited Alzheimer Network (DIAN). *Neuropathology*.

Cao, Shaosheng, Wei Lu, and Qiongkai Xu. 2016. Deep neural networks for learning graph representations. In *Thirtieth aaai conference on artificial intelligence*.

Carroll, J Douglas, and Jih-Jie Chang. 1970. Analysis of individual differences in multidimensional scaling via an n-way generalization of âŁœeckart-youngâŁž decomposition. *Psychometrika* 35(3):283–319.

Chandrasekaran, Arjun, Ashwin K Vijayakumar, Stanislaw Antol, Mohit Bansal, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2016. We are humor beings: Understanding and predicting visual humor. In *Cvpr*.

Chen, Xinlei, and C Lawrence Zitnick. 2015. Mind's eye: A recurrent visual representation for image caption generation. In *Cvpr*.

Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Choi, Wongun, Yu-Wei Chao, Caroline Pantofaru, et al. 2013. Understanding indoor scenes using 3d geometric phrases. In *Cvpr*.

Chua, Terence C, Wei Wen, Melissa J Slavin, and Perminder S Sachdev. 2008. Diffusion tensor imaging in mild cognitive impairment and Alzheimer's disease: a review. *Current opinion in neurology* 21(1):83–92.

Chung, Fan RK, and Fan Chung Graham. 1997. *Spectral graph theory*. 92, American Mathematical Soc.

Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Clark, Lindsay R, Annie M Racine, Rebecca L Koscik, Ozioma C Okonkwo, Corinne D Engelman, Cynthia M Carlsson, Sanjay Asthana, Barbara B Bendlin, Rick Chappell, and Christopher R Nicholas. 2016. Beta-amyloid and cognitive decline in late middle age: findings from the wisconsin registry for Alzheimer's prevention study. *Alzheimer's & Dementia* 12(7): 805–814.

Cohen, Jacob. 2013. *Statistical power analysis for the behavioral sciences*. Routledge.

Collins, Maxwell D, Ji Liu, Jia Xu, et al. 2014. Spectral clustering with a convex regularizer on millions of images. In *Eccv*.

Cook, PA, Y Bai, SKKS Nedjati-Gilani, et al. 2006. Camino: open-source diffusion-MRI reconstruction and processing. In *Ismrm*, vol. 2759.

Corder, EH, AM Saunders, WJ Strittmatter, et al. 1993. Gene dose of apolipoprotein E type 4 allele and the risk of Alzheimer's disease in late onset families. *Science* 261(5123):921–923.

Cricri, Francesco, Mikko Honkala, Xingyang Ni, Emre Aksu, and Moncef Gabbouj. 2016. Video ladder networks. *arXiv preprint arXiv:1612.01756*.

De Brabandere, Bert, Xu Jia, Tinne Tuytelaars, and Luc Van Gool. 2016. Dynamic filter networks. In *Nips*.

Deng, Jia, Nan Ding, Yangqing Jia, et al. 2014. Large-scale object classification using label relation graphs. In *Eccv*.

Der Kiureghian, Armen, and Ove Ditlevsen. 2009. Aleatory or epistemic? does it matter? *Structural Safety* 31(2):105–112.

Desai, Chaitanya, and Deva Ramanan. 2012. Detecting actions, poses, and objects with relational phraselets. In *Eccv*. Springer.

Desikan, Rahul S, Florent Ségonne, Bruce Fischl, et al. 2006. An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *Neuroimage* 31(3):968–980.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: Human language technologies*, 4171–4186.

Dinh, Laurent, David Krueger, and Yoshua Bengio. 2014. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*.

Dinh, Laurent, Jascha Sohl-Dickstein, and Samy Bengio. 2016. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*.

Drzezga, Alexander, J Alex Becker, Koene RA Van Dijk, et al. 2011. Neuronal dysfunction and disconnection of cortical hubs in non-demented subjects with elevated amyloid burden. *Brain* 134(6):1635–1646.

Dziugaite, Gintare Karolina, Daniel M Roy, and Zoubin Ghahramani. 2015. Training generative neural networks via maximum mean discrepancy optimization. *arXiv preprint arXiv:1505.03906*.

Eslami, SM, Nicolas Heess, Theophane Weber, et al. 2016. Attend, Infer, Repeat: Fast Scene Understanding with Generative Models. *arXiv preprint arXiv:1603.08575*.

Esteban, Cristóbal, Oliver Staeck, Stephan Baier, Yinchong Yang, and Volker Tresp. 2016. Predicting clinical events by combining static and dynamic information using recurrent neural networks. In *Ichi*.

Fischer, F. U., D. Wolf, A. Scheurich, A. Fellgiebel, and Initiative Alzheimer's Disease Neuroimaging. 2015. Altered whole-brain white matter networks in preclinical alzheimer's disease. *Neuroimage Clin* 8: 660–6.

Fisher, Ronald A. 1936. The use of multiple measurements in taxonomic problems. *Annals of eugenics* 7(2):179–188.

Fleisher, Adam S, Kewei Chen, Xiaofen Liu, Napatkamon Ayutyanont, Auttawut Roontiva, Pradeep Thiyyagura, Hillary Protas, Abhinay D Joshi, Marwan Sabbagh, Carl H Sadowsky, et al. 2013. Apolipoprotein e ε4 and age effects on florbetapir positron emission tomography in healthy aging and alzheimer disease. *Neurobiology of aging* 34(1):1–12.

Fortunato, Meire, Charles Blundell, and Oriol Vinyals. 2017. Bayesian Recurrent Neural Networks. *arXiv preprint arXiv:1704.02798*.

Fox, NC, EK Warrington, PA Freeborough, P Hartikainen, AM Kennedy, JM Stevens, and Martin N Rossor. 1996. Presymptomatic hippocampal atrophy in alzheimer's disease: A longitudinal mri study. *Brain* 119(6): 2001–2007.

Fraccaro, Marco, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. 2016. Sequential neural models with stochastic layers. In *Advances in neural information processing systems*, 2199–2207.

Friedman, Nir, Lise Getoor, Daphne Koller, et al. 1999. Learning probabilistic relational models. In *Ijcai*.

Gabriel, Karl Ruben. 1971. The biplot graphic display of matrices with application to principal component analysis. *Biometrika* 58(3):453–467.

Gal, Yarin, and Zoubin Ghahramani. 2015. Bayesian convolutional neural networks with Bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*.

———. 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Icml*.

Gers, Felix A, Jürgen Schmidhuber, and Fred Cummins. 1999. Learning to forget: Continual prediction with LSTM.

Getoor, Lise, Nir Friedman, Daphne Koller, et al. 2001. Learning probabilistic models of relational structure. In *Icml*.

Ghosh, Arnab, Viveka Kulharia, Amitabha Mukerjee, Vinay Namboodiri, and Mohit Bansal. 2016. Contextual RNN-GANs for abstract reasoning diagram generation. *arXiv preprint arXiv:1609.09444*.

Gorski, Jochen, Frank Pfeuffer, and Kathrin Klamroth. 2007. Biconvex sets and optimization with biconvex functions: a survey and extensions. *Mathematical Methods of Operations Research* 66.

Graves, Alex. 2011. Practical variational inference for neural networks. In *Nips*.

Gregor, Karol, George Papamakarios, Frederic Besse, Lars Buesing, and Theophane Weber. 2018. Temporal difference variational auto-encoder. *arXiv preprint arXiv:1806.03107*.

Greicius, Michael D, Kaustubh Supekar, Vinod Menon, and Robert F Dougherty. 2009. Resting-state functional connectivity reflects structural connectivity in the default mode network. *Cerebral cortex* 19(1):72–78.

Gretton, Arthur, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *Journal of Machine Learning Research* 13(Mar):723–773.

Guye, M., G. Bettus, F. Bartolomei, and P. J. Cozzone. 2010. Graph theoretical analysis of structural and functional connectivity mri in normal and pathological brain networks. *MAGMA* 23(5-6):409–21.

Ham, Jihun, Daniel Lee, and Lawrence Saul. 2005. Semisupervised alignment of manifolds. In *Uai*, vol. 10, 120–127.

Hardoon, David R, Sandor Szedmak, and John Shawe-Taylor. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural computation* 16(12):2639–2664.

Hardy, J. A., and G. A. Higgins. 1992. Alzheimer's disease: the amyloid cascade hypothesis. *Science* 256(5054):184–5.

Harshman, Richard A, and Margaret E Lundy. 1994. PARAFAC: Parallel factor analysis. *Computational Statistics & Data Analysis* 18.

Harshman, Richard A, et al. 1970. Foundations of the parafac procedure: Models and conditions for an" explanatory" multimodal factor analysis.

Hernández-Lobato, José Miguel, and Ryan Adams. 2015. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *Icml*.

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hinton, Geoffrey E, and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science* 313(5786):504–507.

Hinton, Geoffrey E, and Drew Van Camp. 1993. Keeping the neural networks simple by minimizing the description length of the weights. In *Computational learning theory*.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Hoštalkova, Eva, and Aleš Procházka. 1905. Wavelet signal and image denoising. *signal* 500:2.

Hotelling, H. 1936. Relations between two sets of variates. *Biometrika* 28(3):321–377.

Hsu, Daniel, and Sham M Kakade. 2013. Learning mixtures of spherical gaussians: moment methods and spectral decompositions. In *Iics*.

Hua, Xue, Alex D Leow, Neelroop Parikshak, et al. 2008. Tensor-based morphometry as a neuroimaging biomarker for Alzheimer's disease: an MRI study of 676 AD, MCI, and normal subjects. *NeuroImage* 43(3): 458–469.

Hwang, Seong Jae, Nagesh Adluru, Maxwell D Collins, Sathya N Ravi, Barbara B Bendlin, Sterling C Johnson, and Vikas Singh. 2016. Coupled Harmonic Bases for Longitudinal Characterization of Brain Networks. *Computer Vision and Pattern Recognition*.

Hwang, Seong Jae, Nagesh Adluru, Won Hwa Kim, Sterling C Johnson, Barbara B Bendlin, and Vikas Singh. 2019a. Associations Between Positron Emission Tomography Amyloid Pathology and Diffusion Tensor Imaging Brain Connectivity in Pre-Clinical Alzheimer's Disease. *Brain connectivity* 9.

Hwang, Seong Jae, Maxwell D Collins, Sathya N Ravi, et al. 2015. A Projection Free Method for Generalized Eigenvalue Problem With a Nonsmooth Regularizer. In *Iccv*.

Hwang, Seong Jae, Ronak Mehta, Hyunwoo J. Kim, Sterling C. Johnson, and Vikas Singh. 2019b. Sampling-free Uncertainty Estimation in Gated Recurrent Units with Applications to Normative Modeling in Neuroimaging. In *Uai*, 296.

Hwang, Seong Jae, Sathya N Ravi, Zirui Tao, Hyunwoo J Kim, Maxwell D Collins, and Vikas Singh. 2018. Tensorize, Factorize and Regularize: Robust Visual Relationship Learning. *Computer Vision and Pattern Recognition*.

Hwang, Seong Jae, Zirui Tao, Won Hwa Kim, and Vikas Singh. 2019c. Conditional Recurrent Flow: Conditional Generation of Longitudinal Samples with Applications to Neuroimaging. *International Conference on Computer Vision*.

Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 1125–1134.

Jack, Clifford R, David A Bennett, Kaj Blennow, Maria C Carrillo, Billy Dunn, Samantha Budd Haeberlein, David M Holtzman, William Jagust, Frank Jessen, and Jason Karlawish. 2018. Nia-aa research framework: Toward a biological definition of alzheimer's disease. *Alzheimer's & Dementia* 14(4):535–562.

Jack, Clifford R, David A Bennett, Kaj Blennow, Maria C Carrillo, Howard H Feldman, Giovanni B Frisoni, Harald Hampel, William J Jagust, Keith A Johnson, David S Knopman, et al. 2016. A/t/n: an unbiased

descriptive classification scheme for alzheimer disease biomarkers. *Neurology* 87(5):539–547.

Jack, Clifford R, Heather J Wiste, Stephen D Weigand, David S Knopman, Prashanthi Vemuri, Michelle M Mielke, Val Lowe, Matthew L Senjem, Jeffrey L Gunter, Mary M Machulda, et al. 2015. Age, sex, and apoe $\varepsilon 4$ effects on memory, brain structure, and β-amyloid across the adult life span. *JAMA neurology* 72(5):511–519.

Jagannatha, Abhyuday N, and Hong Yu. 2016. Bidirectional rnn for medical event detection in electronic health records. In *Association for computational linguistics*.

Jain, Prateek, Raghu Meka, and Inderjit S Dhillon. 2010. Guaranteed rank minimization via singular value projection. In *Nips*.

Jbabdi, Saad, Stamatios N Sotiropoulos, Suzanne N Haber, et al. 2015. Measuring macroscopic brain connections in vivo. *Nature neuroscience* 18(11):1546–1555.

Jenatton, Rodolphe, Nicolas L Roux, Antoine Bordes, and Guillaume R Obozinski. 2012. A latent factor model for highly multi-relational data. In *Advances in neural information processing systems*, 3167–3175.

Jin, Kunlin, Alyson L Peel, Xiao Ou Mao, Lin Xie, Barbara A Cottrell, David C Henshall, and David A Greenberg. 2004. Increased hippocampal neurogenesis in alzheimer's disease. *Proceedings of the National Academy of Sciences* 101(1):343–347.

Johnson, Justin, Ranjay Krishna, Michael Stark, et al. 2015. Image retrieval using scene graphs. In *Cvpr*.

Johnson, S. C., B. T. Christian, O. C. Okonkwo, J. M. Oh, S. Harding, G. Xu, A. T. Hillmer, D. W. Wooten, D. Murali, T. E. Barnhart, L. T. Hall, A. M.

Racine, W. E. Klunk, C. A. Mathis, B. B. Bendlin, C. L. Gallagher, C. M. Carlsson, H. A. Rowley, B. P. Hermann, N. M. Dowling, S. Asthana, and M. A. Sager. 2014a. Amyloid burden and neural function in people at risk for alzheimer's disease. *Neurobiol Aging* 35(3):576–84.

Johnson, Sterling C, Bradley T Christian, Ozioma C Okonkwo, Jennifer M Oh, Sandra Harding, Guofan Xu, Ansel T Hillmer, Dustin W Wooten, Dhanabalan Murali, Todd E Barnhart, Lance T Hall, Annie M Racine, William E Klunk, Chester A Mathis, Howard A Rowley, Bruce P Hermann, N. Maritza Dowling, Sanjay Asthana, and Mark A Sager. 2014b. Amyloid burden and neural function in people at risk for Alzheimer's disease. *Neurobiology of aging* 35(3):576–584.

Joshi, Abhinay D, Michael J Pontecorvo, Chrisopher M Clark, et al. 2012. Performance characteristics of amyloid PET with florbetapir f 18 in patients with alzheimer's disease and cognitively normal subjects. *Journal of Nuclear Medicine* 53(3):378–384.

Jozefowicz, Rafal, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Jozefowicz, Rafal, Wojciech Zaremba, and Ilya Sutskever. 2015. An empirical exploration of recurrent network architectures. In *International conference on machine learning*, 2342–2350.

Keihaninejad, Shiva, Hui Zhang, Natalie S Ryan, et al. 2013. An unbiased longitudinal analysis framework for tracking white matter changes using diffusion tensor imaging with application to Alzheimer's disease. *NeuroImage* 72:153–163.

Kendall, Alex, and Yarin Gal. 2017. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, 5574–5584.

Khachaturian, Ara S, Christopher D Corcoran, Lawrence S Mayer, Peter P Zandi, and John CS Breitner. 2004. Apolipoprotein e $\varepsilon 4$ count affects age at onset of alzheimer disease, but not lifetime susceptibility: the cache county study. *Archives of general psychiatry* 61(5):518–524.

Kim, Won Hwa, Nagesh Adluru, Moo K Chung, Ozioma C Okonkwo, Sterling C Johnson, Barbara Bendlin, and Vikas Singh. 2015. Multi-resolution statistical analysis of brain connectivity graphs in preclinical Alzheimer's disease. *NeuroImage*.

Kim, Won Hwa, Moo K Chung, and Vikas Singh. 2013. Multi-resolution shape analysis via non-Euclidean wavelets: Applications to mesh segmentation and surface alignment problems. In *Cvpr*, 2139–2146. IEEE.

Kim, Won Hwa, Annie M Racine, Nagesh Adluru, Seong Jae Hwang, Kaj Blennow, Henrik Zetterberg, Cynthia M Carlsson, Sanjay Asthana, Rebecca L Koscik, Sterling C Johnson, Barbara B Bendlin, and Vikas Singh. 2018. Cerebrospinal fluid biomarkers of neurofibrillary tangles and synaptic dysfunction are associated with longitudinal decline in white matter connectivity: A multi-resolution graph analysis. *NeuroImage: Clinical*.

Kim, Won Hwa, Annie M Racine, Nagesh Adluru, Seong Jae Hwang, et al. 2019. Cerebrospinal fluid biomarkers of neurofibrillary tangles and synaptic dysfunction are associated with longitudinal decline in white matter connectivity: A multi-resolution graph analysis. *NeuroImage: Clinical* 21.

Kingma, Diederik P, and Prafulla Dhariwal. 2018. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*.

Kingma, Diederik P, Tim Salimans, and Max Welling. 2015. Variational dropout and the local reparameterization trick. In *Nips*.

Kolda, Tamara G, and Brett W Bader. 2009. Tensor decompositions and applications. *SIAM review* 51(3):455–500.

Koscik, Rebecca, Tobey J Betthauser, Erin M Jonaitis, Lindsay R Clark, Samantha Allison, Kimberly D Mueller, Bruce P Hermann, Jennifer D Poetter, Leah Sanson, Heather Shouel, Nathaniel A Chin, Bradley T Christian, and Sterling C Johnson. 2019a. Modeling pib pet trajectory groups identifies a subgroup with pib beta-amyloid accumulation near age 50 and predicts mk-6240 suvr. In *Human amyloid imaging*.

Koscik, Rebecca L, Tobey J Betthauser, Erin M Jonaitis, Samantha L Allison, Lindsay R Clark, Bruce P Hermann, Karly A Cody, Jonathan W Engle, Todd E Barnhart, Charles K Stone, et al. 2019b. Amyloid duration is associated with preclinical cognitive decline and tau pet. *BioRxiv* 778415.

Kovnatsky, Artiom, Michael M Bronstein, Alexander M Bronstein, et al. 2013. Coupled quasi-harmonic bases. In *Computer graphics forum*, vol. 32, 439–448. Wiley Online Library.

Krishna, Ranjay, Yuke Zhu, Oliver Groth, et al. 2016. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. In *Ijcv*.

Krishnan, Rahul G, Uri Shalit, and David Sontag. 2017. Structured Inference Networks for Nonlinear State Space Models. In *Aaai*, 2101–2109.

Krizhevsky, Alex, and Geoffrey E Hinton. 2011. Using very deep autoencoders for content-based image retrieval. In *Esann*, vol. 1, 2.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Nips*.

Kulchandani, Jaya S, and Kruti J Dangarwala. 2015. Moving object detection: Review of recent research trends. In *Icpc*. IEEE.

Ladicky, Lubor, Chris Russell, Pushmeet Kohli, et al. 2010. Graph cut based inference with co-occurrence statistics. In *Eccv*.

Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell. 2016. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. *arXiv preprint arXiv:1612.01474*.

Lambert, Jonathan, Peter B Greer, Fred Menk, Jackie Patterson, Joel Parker, Kara Dahl, Sanjiv Gupta, Anne Capp, Chris Wratten, Colin Tang, et al. 2011. MRI-guided prostate radiation therapy planning: Investigation of dosimetric accuracy of MRI-based dose planning. *Radiotherapy and Oncology* 98(3):330–334.

Landin-Romero, Ramon, Fiona Kumfor, Cristian E Leyton, Muireann Irish, John R Hodges, and Olivier Piguet. 2017. Disease-specific patterns of cortical and subcortical degeneration in a longitudinal study of alzheimer's disease and behavioural-variant frontotemporal dementia. *Neuroimage* 151:72–80.

Lawrence, Steve, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. 1997. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks* 8(1):98–113.

Lebeda, Karel, Simon Hadfield, and Richard Bowden. 2015. Exploring Causal Relationships in Visual Object Tracking. In *Iccv*.

Lei, Zhen, and Stan Z Li. 2009. Coupled spectral regression for matching heterogeneous faces. In *Cvpr*.

Leibig, Christian, Vaneeda Allken, Murat Seçkin Ayhan, Philipp Berens, and Siegfried Wahl. 2017. Leveraging uncertainty information from deep neural networks for disease detection. *Scientific reports* 7(1):17816.

Li, Chun-Liang, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. 2017a. Mmd gan: Towards deeper understanding of moment

matching network. In *Advances in neural information processing systems*, 2203–2213.

Li, Jerry, Aleksander Madry, John Peebles, and Ludwig Schmidt. 2017b. Towards understanding the dynamics of generative adversarial networks. *arXiv preprint arXiv:1706.09884*.

Li, Weixin, Jungseock Joo, Hang Qi, et al. 2016. Joint Image-Text News Topic Detection and Tracking by Multimodal Topic And-Or Graph. *Multimedia*.

Li, Yujia, Kevin Swersky, and Rich Zemel. 2015. Generative moment matching networks. In *International conference on machine learning*, 1718–1727.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Liu, Ji, Stephen J Wright, Christopher Ré, et al. 2015. An asynchronous parallel stochastic coordinate descent algorithm. *JMLR* 16(1):285–322.

Lu, Canyi, Jinhui Tang, Shuicheng Yan, et al. 2014a. Generalized nonconvex nonsmooth low-rank minimization. In *Cvpr*.

Lu, Cewu, Ranjay Krishna, Michael Bernstein, et al. 2016. Visual relationship detection with language priors. In *Eccv*.

Lu, Yang, Tianfu Wu, and Song Chun Zhu. 2014b. Online object tracking, learning and parsing with and-or graphs. In *Cvpr*.

Ma, Chao, Jun Wang, Junying Zhang, Kewei Chen, Xin Li, Ni Shu, Yaojing Chen, Zhen Liu, and Zhanjun Zhang. 2017. Disrupted brain structural

connectivity: Pathological interactions between genetic apoe ϵ 4 status and developed mci condition. *Molecular neurobiology* 54(9):6999–7007.

MacKay, David JC. 1992a. Bayesian methods for adaptive models. Ph.D. thesis, California Institute of Technology.

———. 1992b. A practical Bayesian framework for backpropagation networks. *Neural computation* 4(3):448–472.

Marinescu, Razvan V, Neil P Oxtoby, Alexandra L Young, et al. 2018. Tadpole challenge: Prediction of longitudinal evolution in alzheimer's disease. *arXiv preprint arXiv:1805.03909*.

Marquand, Andre F, Iead Rezek, Jan Buitelaar, and Christian F Beckmann. 2016. Understanding heterogeneity in clinical cohorts using normative models: beyond case-control studies. *Biological psychiatry* 80(7):552–561.

McCulloch, Warren S, and Walter Pitts. 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* 5(4):115–133.

McLachlan, Geoffrey. 2004. *Discriminant analysis and statistical pattern recognition*, vol. 544. John Wiley & Sons.

Mensink, Thomas, Efstratios Gavves, and Cees GM Snoek. 2014. Costa: Co-occurrence statistics for zero-shot classification. In *Cvpr*.

Miech, Antoine, Ivan Laptev, and Josef Sivic. 2017. Learnable pooling with Context Gating for video classification. *arXiv preprint arXiv:1706.06905*.

Mikolov, Tomas, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*, vol. 2, 3.

Milgrom, Paul, and Ilya Segal. 2002. Envelope theorems for arbitrary choice sets. *Econometrica* 70(2):583–601.

Mirza, Mehdi, and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Monti, Federico, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. 2017. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 5115–5124.

Myshkov, Pavel, and Simon Julier. 2016. Posterior distribution analysis for bayesian inference in neural networks. *Advances in Neural Information Processing Systems (NIPS)*.

Nair, Tanya, Doina Precup, Douglas L Arnold, and Tal Arbel. 2018. Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation. In *International conference on medical image computing and computer-assisted intervention*, 655–663. Springer.

Naj, Adam C, Gyungah Jun, Christiane Reitz, Brian W Kunkle, William Perry, Yo Son Park, Gary W Beecham, Ruchita A Rajbhandary, Kara L Hamilton-Nelson, Li-San Wang, et al. 2014. Effects of multiple genetic loci on age at onset in late-onset alzheimer disease: a genome-wide association study. *JAMA neurology* 71(11):1394–1404.

Nickel, Maximilian, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Icml*.

Nocedal, J., and S. J. Wright. 2006. *Numerical optimization*. 2nd ed. New York: Springer.

Ossenkoppele, Rik, Marissa D Zwan, Nelleke Tolboom, Danielle ME van Assema, Sofie F Adriaanse, Reina W Kloet, Ronald Boellaard, Albert D Windhorst, Frederik Barkhof, Adriaan A Lammertsma, et al. 2012. Amyloid burden and metabolic function in early-onset alzheimer's disease: parietal lobe involvement. *Brain* 135(7):2115–2125.

Papalexakis, Evangelos E, Leman Akoglu, and Dino Ience. 2013. Do more views of a graph help? community detection and clustering in multigraphs. In *Proceedings of the 16th international conference on information fusion*, 899–905. IEEE.

Papamakarios, George, and Iain Murray. 2016. Fast ε-free inference of simulation models with bayesian conditional density estimation. In *Nips*, 1028–1036.

Parlett, Beresford N. 1998. *The symmetric eigenvalue problem*, vol. 20. siam.

Patel, Vishal M, Raghuraman Gopalan, Ruonan Li, et al. 2015. Visual domain adaptation: A survey of recent advances. *Signal Processing Magazine* 32(3):53–69.

Pearson, Karl. 1901. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2(11):559–572.

Pike, K. E., G. Savage, V. L. Villemagne, S. Ng, S. A. Moss, P. Maruff, C. A. Mathis, W. E. Klunk, C. L. Masters, and C. C. Rowe. 2007. Beta-amyloid imaging and memory in non-demented individuals: evidence for preclinical alzheimer's disease. *Brain* 130(Pt 11):2837–44.

Racine, Annie M, Nagesh Adluru, Andrew L Alexander, Bradley T Christian, Ozioma C Okonkwo, Jennifer Oh, Caitlin A Cleary, Alex Birdsill, Ansel T Hillmer, and Dhanabalan Murali. 2014. Associations between white matter microstructure and amyloid burden in preclinical alzheimer's disease: a multimodal imaging investigation. *NeuroImage: Clinical* 4:604–614.

Racine, Annie M, Lindsay R Clark, Sara E Berman, Rebecca L Koscik, Kimberly D Mueller, Derek Norton, Christopher R Nicholas, Kaj Blennow,

Henrik Zetterberg, Bruno Jedynak, et al. 2016. Associations between performance on an abbreviated cogstate battery, other measures of cognitive function, and biomarkers in people at risk for Alzheimer's disease. *Journal of Alzheimer's Disease* 54(4):1395–1408.

Raj, Ashish, Amy Kuceyeski, and Michael Weiner. 2012. A network diffusion model of disease progression in dementia. *Neuron* 73(6):1204–1215.

Ramanathan, Vignesh, Congcong Li, Jia Deng, et al. 2015. Learning semantic relationships for better action retrieval in images. In *Cvpr*.

Ranganath, Rajesh, Linpeng Tang, Laurent Charlin, and David Blei. 2015. Deep exponential families. In *Artificial intelligence and statistics*, 762–771.

Rasmussen, Carl Edward, and Joaquin Quinonero-Candela. 2005. Healing the relevance vector machine through augmentation. In *Icml*.

Reijneveld, Jaap C, Sophie C Ponten, Henk W Berendse, et al. 2007. The application of graph theoretical analysis to complex networks in the brain. *Clinical Neurophysiology* 118(11):2317–2331.

Reiman, Eric M, Kewei Chen, Xiaofen Liu, et al. 2009. Fibrillar amyloid-β burden in cognitively normal people at 3 levels of genetic risk for Alzheimer's disease. *Proceedings of the National Academy of Sciences* 106(16):6820–6825.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Nips*.

Rezende, Danilo Jimenez, and Shakir Mohamed. 2015. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*.

Ribeiro, Fabio De Sousa, Francesco Caliva, Mark Swainson, Kjartan Gudmundsson, Georgios Leontidis, and Stefanos Kollias. 2018. Deep bayesian uncertainty estimation for adaptation and self-annotation of food packaging images. *arXiv preprint arXiv:1812.01681*.

Richardson, Matthew, and Pedro Domingos. 2006. Markov logic networks. *Machine learning* 62.

Rippel, Oren, and Ryan Prescott Adams. 2013. High-dimensional probability estimation with deep density models. *arXiv preprint arXiv:1302.5125*.

Rohrbach, Marcus, Wei Qiu, Ivan Titov, et al. 2013. Translating video content to natural language descriptions. In *Iccv*.

Rosenberg, Samuel J, Joseph J Ryan, and Aurelio Prifitera. 1984. Rey auditory-verbal learning test performance of patients with and without memory impairment. *Journal of clinical psychology* 40(3):785–787.

Rosenblatt, Frank. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review* 65(6): 386.

Roth, Kevin, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. 2017. Stabilizing training of generative adversarial networks through regularization. In *Advances in neural information processing systems*, 2018–2028.

Rubinov, Mikail, and Olaf Sporns. 2010. Complex network measures of brain connectivity: uses and interpretations. *NeuroImage* 52(3):1059–1069.

Saad, Youcef. 1992. *Numerical methods for large eigenvalue problems*, vol. 158. SIAM.

Sadeghi, Mohammad Amin, and Ali Farhadi. 2011. Recognition using visual phrases. In *Cvpr*.

Sak, Haşim, Andrew Senior, and Françoise Beaufays. 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth annual conference of the international speech communication association*.

Sanghavi, Sujay, Rachel Ward, and Chris D White. 2017. The local convexity of solving systems of quadratic equations. *Results in Mathematics* 71(3-4):569–608.

Santeramo, Ruggiero, Samuel Withey, and Giovanni Montana. 2018. Longitudinal detection of radiological abnormalities with time-modulated lstm. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, 326–333.

Schmidt, Michael, et al. 1996. *Rey auditory verbal learning test: a handbook*. Western Psychological Services Los Angeles.

Sedlmeier, Andreas, Thomas Gabor, Thomy Phan, Lenz Belzner, and Claudia Linnhoff-Popien. 2019. Uncertainty-based out-of-distribution detection in deep reinforcement learning. *arXiv preprint arXiv:1901.02219*.

Shi, Jianbo, and Jitendra Malik. 2000. Normalized cuts and image segmentation. *Departmental Papers (CIS)* 107.

Singh, Ajit P, and Geoffrey J Gordon. 2008. Relational learning via collective matrix factorization. In *Acm sigkdd*. ACM.

Sohn, Kihyuk, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. In *Nips*, 3483–3491.

Sojkova, Jitka, Yun Zhou, Yang An, Michael A Kraut, Luigi Ferrucci, Dean F Wong, and Susan M Resnick. 2011. Longitudinal patterns of β-amyloid deposition in nondemented older adults. *Archives of neurology* 68(5):644–649.

Song, Young Chol, Henry Kautz, James Allen, et al. 2013. A markov logic framework for recognizing complex events from multimodal data. In *International conference on multimodal interaction*. ACM.

Sotiropoulos, Stamatios N, Saad Jbabdi, Junqian Xu, et al. 2013. Advances in diffusion MRI acquisition and processing in the Human Connectome Project. *NeuroImage* 80:125–143.

Sperling, Reisa A, Paul S Aisen, Laurel A Beckett, David A Bennett, Suzanne Craft, Anne M Fagan, Takeshi Iwatsubo, Clifford R Jack Jr, Jeffrey Kaye, Thomas J Montine, et al. 2011. Toward defining the preclinical stages of Alzheimer's disease: Recommendations from the National Institute on Aging-Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease. *Alzheimer's & dementia* 7(3):280–292.

Sporns, Olaf. 2011. The human connectome: a complex network. *Annals of the New York Academy of Sciences* 1224(1):109–125.

Sprecher, Kate E, Barbara B Bendlin, Annie M Racine, Ozioma C Okonkwo, Bradley T Christian, Rebecca L Koscik, Mark A Sager, Sanjay Asthana, Sterling C Johnson, and Ruth M Benca. 2015. Amyloid burden is associated with self-reported sleep in nondemented late middle-aged adults. *Neurobiology of aging* 36(9):2568–2576.

Sriperumbudur, Bharath K, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert RG Lanckriet. 2009. On integral probability metrics,\phi-divergences and binary classification. *arXiv preprint arXiv:0901.2698*.

Srivastava, Nitish, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *JMLR* 15(1):1929–1958.

Srivastava, Nitish, Elman Mansimov, and Ruslan Salakhudinov. 2015. Unsupervised learning of video representations using lstms. In *Icml*.

Stam, Cornelis J, and Jaap C Reijneveld. 2007. Graph theoretical analysis of complex networks in the brain. *Nonlinear biomedical physics* 1(1):3.

Sun, Ruoyu, and Zhi-Quan Luo. 2015. Guaranteed matrix completion via nonconvex factorization. In *Focs*.

Tang, Duyu, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, 1422–1432.

Thambisetty, Madhav, Yang An, and Toshiko Tanaka. 2013. Alzheimer's disease risk genes and the age-at-onset phenotype. *Neurobiology of aging* 34(11):2696–e1.

Thomason, Jesse, Subhashini Venugopalan, Sergio Guadarrama, et al. 2014. Integrating Language and Vision to Generate Natural Language Descriptions of Videos in the Wild. In *Coling*.

Tolstikhin, Ilya, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. 2017. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*.

Tombaugh, Tom N, and Nancy J McIntyre. 1992. The mini-mental state examination: a comprehensive review. *Journal of the American Geriatrics Society* 40(9):922–935.

Tran, Son D, and Larry S Davis. 2008. Event modeling and recognition using markov logic networks. In *Eccv*. Springer.

Tropp, Joel A. 2015. An introduction to matrix concentration inequalities. *Found. Trends Mach. Learn.*

Tu, Stephen, Ross Boczar, Mahdi Soltanolkotabi, et al. 2015. Low-rank solutions of linear matrix equations via procrustes flow. *arXiv preprint arXiv:1507.03566*.

Tucker, Ledyard R. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika* 31.

Turk, Matthew A, and Alex P Pentland. 1991. Face recognition using eigenfaces. In *Cvpr*. IEEE.

Tzourio-Mazoyer, N., B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot. 2002. Automated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the mni mri single-subject brain. *Neuroimage* 15(1):273–89.

Uğurbil, Kamil, Junqian Xu, Edward J Auerbach, et al. 2013. Pushing spatial and temporal resolution for functional and diffusion MRI in the Human Connectome Project. *NeuroImage* 80:80–104.

Van Essen, David C, Kamil Ugurbil, E Auerbach, et al. 2012. The Human Connectome Project: a data acquisition perspective. *NeuroImage* 62(4): 2222–2231.

Varentsova, Anna, Shengwei Zhang, and Konstantinos Arfanakis. 2014. Development of a high angular resolution diffusion imaging human brain template. *NeuroImage* 91:177–186.

Villemagne, Victor L, Samantha Burnham, Pierrick Bourgeat, Belinda Brown, Kathryn A Ellis, Olivier Salvado, Cassandra Szoeke, S Lance Macaulay, Ralph Martins, Paul Maruff, et al. 2013. Amyloid β deposition, neurodegeneration, and cognitive decline in sporadic alzheimer's disease: a prospective cohort study. *The Lancet Neurology* 12(4):357–367.

Vlassenko, Andrei G, Mark A Mintun, Chengjie Xiong, Yvette I Sheline, Alison M Goate, Tammie LS Benzinger, and John C Morris. 2011. Amyloid-beta plaque growth in cognitively normal adults: Longitudinal [11c] pittsburgh compound b data. *Annals of neurology* 70(5):857–861.

Vu, Van. 2008. Random discrete matrices. In *Horizons of combinatorics*, 257–280. Springer.

Wang, Chang, and Sridhar Mahadevan. 2008. Manifold alignment using Procrustes analysis. In *Icml*.

Wang, Hao, Shi Xingjian, and Dit-Yan Yeung. 2016a. Natural-parameter networks: A class of probabilistic neural networks. In *Nips*.

Wang, Jingya, Mohammed Korayem, Saul Blanco, et al. 2016b. Tracking Natural Events through Social Media and Computer Vision. In *Multimedia*. ACM.

Wang, Pan, Bo Zhou, Hongxiang Yao, et al. 2015. Aberrant intra-and inter-network connectivity architectures in Alzheimer's disease and mild cognitive impairment. *Scientific reports* 5.

Warsa, James S, Todd A Wareing, Jim E Morel, et al. 2004. Krylov subspace iterations for deterministic k-eigenvalue calculations. *Nuclear Science and Engineering* 147(1):26–42.

Welling, Max, and Yee W Teh. 2011. Bayesian learning via stochastic gradient Langevin dynamics. In *Icml*.

Wen, Zaiwen, and Wotao Yin. 2013. A feasible method for optimization with orthogonality constraints. *Mathematical Programming* 142(1-2):397–434.

Werbos, Paul. 1974. Beyond regression:" new tools for prediction and analysis in the behavioral sciences. *Ph. D. dissertation, Harvard University*.

Wig, Gagan S, Bradley L Schlaggar, and Steven E Petersen. 2011. Concepts and principles in the analysis of brain networks. *Annals of the New York Academy of Sciences* 1224(1):126–146.

Wilkinson, James Hardy. 1965. *The algebraic eigenvalue problem*, vol. 662. Oxford Clarendon.

Wong, Dean F, Paul B Rosenberg, Yun Zhou, et al. 2010. In vivo imaging of Amyloid deposition in Alzheimer's disease using the novel radioligand [18f] av-45 (florbetapir f 18). *Journal of nuclear medicine* 51(6):913.

Wu, Anqi, Sebastian Nowozin, Edward Meeds, Richard E Turner, José Miguel Hernández-Lobato, and Alexander L Gaunt. 2019. Deterministic variational inference for robust bayesian neural networks. In *Iclr*.

Wu, Qi, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. 2016a. Ask me anything: Free-form visual question answering based on knowledge from external sources. In *Cvpr*.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016b. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xiao, Han, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xiao, Yijun, and William Yang Wang. 2019. Quantifying uncertainties in natural language processing tasks. In *Proceedings of the aaai conference on artificial intelligence*, vol. 33, 7322–7329.

Xu, Danfei, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. 2017. Scene graph generation by iterative message passing. In *Cvpr*.

Xu, Kelvin, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Icml*.

Yao, Bangpeng, and Li Fei-Fei. 2010. Modeling mutual context of object and human pose in human-object interaction activities. In *Cvpr*.

Zhan, Yibing, Jun Yu, Ting Yu, and Dacheng Tao. 2019. On exploring undetermined relationships for visual relationship detection. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 5128–5137.

Zhang, Hui, Brian B Avants, Paul A Yushkevich, John H Woo, Sumei Wang, Leo F McCluskey, Lauren B Elman, Elias R Melhem, and James C Gee. 2007. High-dimensional spatial normalization of diffusion tensor images improves the detection of white matter differences: an example study using amyotrophic lateral sclerosis. *Medical Imaging, IEEE Transactions on* 26(11):1585–1597.

Zhao, Rui, Dongzhe Wang, Ruqiang Yan, Kezhi Mao, Fei Shen, and Jinjiang Wang. 2017. Machine health monitoring using local feature-based gated recurrent unit networks. *IEEE Transactions on Industrial Electronics* 65(2):1539–1548.

Zhao, Wenyi, Rama Chellappa, and P Jonathon Phillips. 1999. *Subspace linear discriminant analysis for face recognition*. Citeseer.

Zhu, Yuke, Oliver Groth, Michael Bernstein, and Li Fei-Fei. 2016. Visual7W: Grounded Question Answering in Images. In *Cvpr*.

Ziegler, Gabriel, Gerard R Ridgway, Robert Dahnke, Christian Gaser, and Alzheimer's Disease Neuroimaging Initiative. 2014. Individualized

Gaussian process-based prediction and detection of local and global gray matter abnormalities in elderly subjects. *NeuroImage* 97:333–348.

Zitnick, C Lawrence, Ramakrishna Vedantam, and Devi Parikh. 2016. Adopting abstract images for semantic scene understanding. *PAMI* 38(4): 627–638.