KERNEL METHODS IN THE ANALYSIS OF BIG AND COMPLEX DATA: A MODERN STATISTICAL CHALLENGE

by

Hao Henry Zhou

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

Doctor of Philosophy

(Statistics)

at the

UNIVERSITY OF WISCONSIN–MADISON

2019

Date of final oral examination: 02/20/2019

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

Grace Wahba, Professor, Statistics, Computer Sciences
Vikas Singh, Professor, Biostatistics & Medical Informatics, Computer Sciences
Garvesh Raskutti, Assistant Professor, Statistics
Anru Zhang, Assistant Professor, Statistics
Hyunseung Kang, Assistant Professor, Statistics

© Copyright by Hao Henry Zhou 2019
All Rights Reserved

ACKNOWLEDGMENTS

It is a memorable journey to complete my Ph.D. program including this dissertation. I was fortunate to have a great privilege and an honor to work on projects that I am passionate about with great mentors, collaborators, and colleagues. I use this space to thank the people who provide invaluable support and help for me on the journey.

First and foremost, I want to express my deepest gratitude to my advisor, Professor Grace Wahba. Without her guidance and encouragement, the research in this dissertation would not be possible. Grace is a distinguished statistician, her passion and insight for statistics and machine learning motivate me to work on original research in pursuit of knowledge and excellence. Grace is a supportive and enlightening advisor, she gave me the freedom for choosing research topics and always shared her invaluable experience with me to help me solve problems when I had difficulties with my research. It is a great honor and privilege to work closely and learn from her during my Ph.D. study, that makes me become a good researcher.

I want to thank Professor Vikas Singh. In my second year, Grace and I worked on building machine learning models to develop people's understanding of Alzheimer's disease. I found a paper on this topic from Vikas and we scheduled a meet to discuss domain adaptation for Alzheimer's disease research. Since then, Vikas, Grace and I have successful collaborations on developing statistical and machine learning methods for pooling data from multiple heterogeneous sources and increasing understanding of Alzheimer's disease with more samples. Vikas became my co-advisor and I learned a lot from him. He taught me how to write a paper, do a good presentation, communicate with collaborators from other fields, and so on. I used to write a draft of a lot of technical details and little discussions. Then, before my graduation, I have published five papers and I provide comments for papers as a reviewer. It is always fun and inspiring to discuss with Vikas about the research projects. I also want to thank Professor Garvesh Raskutti. In my third year, I looked for working on the fundamental theory of statistics. He gave me a topic on

'high-dimensional multivariate time series'. Original research on statistical theory can make people frustrated when the proof seems unable to continue. Garvesh is patient, supportive and enlightening in helping me solve these problems. This successful collaboration helps me build up my confidence in that when I met hard problems, I can always find ways to solve them.

I want to thank Professor Sterling C. Johnson, Professor Ming Yuan, Professor Anru Zhang, and Professor Hyunseung Kang. For the applications of my research on Alzheimer's disease, Professor Johnson provided precious data for us and helped us interpret the results as an expert on the Alzheimer's disease study. When I first came to UW-Madison, Professor Yuan shared his experience and insight with me helping me understand how to become an independent researcher and work on original research. I want to thank Professor Zhang and Professor Kang for serving on my committee and give me great advice and comments.

I want to thank Vamsi Ithapu, Yunyang Xiong, Sathya Ravi and Ronak Mehta for their effort on our collaborative projects. Thanks all the students from Thursday group meetings, including Jing Kong, Tai Qin, Luwan Zhang, Han Chen, Shulei Wang, Cuize Han, Yilin Zhang, Xiaowu Dai, Lili Zheng and Yuchen Zhou for presenting the works and asking questions for my presentation.

Lastly, I want to thank my family for unconditioned and constant support. Without my parents' support, I could not be able to even start this journey. Thank Yilin Zhang for the support and understanding during my Ph.D. study. We have made the important decisions together, cheered each other up on bad days, and shared all those unforgettable and precious memories over the past five years.

My research during the Ph.D. study is supported by the Center for Predictive Computational Phenotyping (CPCP), which develops innovative computational and statistical methods and software for a broad range of problems that can be cast as computational phenotyping. It helped Grace, Vikas and me to find the opportunity for collaboration. My work is also supported by NIH R01AG040396-01A1 and NSF 1308877.

TABLE OF CONTENTS

List of Tables **v**

List of Figures **vi**

Abstract **viii**

1 Introduction **1**

1.1 Kernel methods 1

1.2 Data is from multiple heterogeneous sources 2

1.3 Data is spatiotemporal and high-dimensional 4

1.4 Data is huge and complex 5

2 Maximum Mean discrepancy and Graphical Causal Model to Pool and Analyze Datasets from Multiple Heterogeneous Sources **7**

2.1 Introduction 7

2.2 Problem Setting 10

2.3 Identifiability Condition 12

2.4 Tests for Correcting Distributional Shift 15

2.5 Experiments 24

2.6 Discussion 31

3 Hypothesis Tests and Theoretical Analysis on When Pooling Multi-Source Datasets is Beneficial **33**

3.1 Introduction 33

3.2 Hypothesis Test for Multi-Site Regression 37

3.3 Pooling in High Dimensional Regression 42

3.4 Experiments 47

3.5 Discussion 54

4 Non-parametric Sparse Additive Auto-regressive Network Models and Theoretical Analysis for Spatio-temporal Data **55**
4.1 Introduction . . . 55
4.2 Preliminaries . . . 58
4.3 Model and estimator . . . 61
4.4 Main results . . . 65
4.5 Proof for the main result (Theorem 4.3) . . . 71
4.6 Numerical experiments . . . 81

5 Understand the Flow of Information in Deep Probabilistic Models: Extend Statistical Models to Deep Structure **89**
5.1 Introduction . . . 89
5.2 Deep Probabilistic Models with a Computation Skeleton Assumption . . . 95
5.3 Additive Structure and Interactions . . . 102
5.4 Experiments . . . 105
5.5 Discussion . . . 108

6 Concluding Remarks **109**

A Appendix **111**
A.1 Proofs for Theorems in Chapter 2 . . . 111
A.2 Proofs for Theorems in Chapter 3 . . . 124
A.3 Proofs for Theorems in Chapter 4 . . . 141
A.4 Supplement for Chapter 5 . . . 167

References **190**

LIST OF TABLES

2.1 Variations of age and diagnosis status across ADNI and W-ADRC . . . 28
2.2 The performance of thresholds in ADNI and W-ADRC. 30

3.1 The detected sparse patterns for the sparse multi-site Lasso model (adding multi-sites Lasso on Lasso) 52
3.2 The detected sparse patterns for the sparse multi-site Lasso model (adding Lasso on multi-sites Lasso) 52

5.1 (L) Synthetic functions from Tsang et al. (2018); (R) Comparisons between BANN, BNN, BART and NID on accuracy and interaction detection 105
5.2 Constructing multiple types BANN of f_1. 106
5.3 Compare BANN with previous works on benchmarks 108

LIST OF FIGURES

2.1 Distributional shift of proteins across ADNI and W-ADRC 10
2.2 Examples for graphical causal model 13
2.3 Searching algorithms for a minimal d-separating set in a graphical causal model 14
2.4 The accuracy of our algorithms to eliminate distributional shift and the acceptance rates of our hypothesis tests for the effectness of the algorithm 25
2.5 The distributions of proteins across ADNI and W-ADRC are aligned well after applying our algorithm 29
2.6 The performance of prediction and classification tasks after pooling ADNI and W-ADRC with our algorithms 32

3.1 After pooling two datasets, the estimation of parameters has higher bias and lower variance. The bias-variance tradeoff prefers the reduction in variance so the estimation improves 38
3.2 The confounding variables should continue to be considered as independent after pooling 41
3.3 The plots (a,d,d,e) shows the performance of our hypothesis tests on when pooling multi-source datasets is beneficial. The plots (c,f) shows that our approach for selecting tuning parameter α outperforms previous approaches in sparse multi-task Lasso 49
3.4 The performance of our hypothesis tests on real datasets of ADNI and W-ADRC (local) 50
3.5 Extra experiment results 53
3.6 Examples for showing how to choose α in our approach 54

4.1 The simulations support our theoretical bound on T and d (bounded noise) 83
4.2 The simulations support our theoretical bound on T and d (Poisson) . 84

4.3 The performance of SpAM on Chicago crime data (boxplot for all communities) 86
4.4 The performance of SpAM on Chicago crime data (community 34 and 56) 87
4.5 Clustering on Chicago crime data based on SpAM 88

5.1 Constructing (d) with (a,b,c) represents our computation skeleton framework to represent BNNs. (e,f,g,h) shows various deep probabilstic models and computation skeletons 95
5.2 Interaction between x_1, x_2 for f_1 in Tab. 5.1. Mean interaction (L) and its standard deviation (R) shown. 107

A.1 BANN learns the additive structure of the function f_1 189

ABSTRACT

Kernel methods have achieved a lot of success in statistics and machine learning where linear models are often not sufficient to capture the relations and patterns in the data. For example, in smoothing spline models, a kernel is used to define a non-parametric class of the function so that one can fit a smoothing curve or surface to understand the relation between the response and predictors. Similarly, in Gaussian process, a kernel is used to define the covariance matrix so that one can understand the nonlinear trend of the data in a Bayesian manner. In support vector machine, a kernel is used to define a non-linear map to transform the predictors into a high-dimensional space so that one can find a hyperplane in the new space to distinguish data from different classes. All these methods over the past two decades have been successful in applications for biomedical science, finance, pattern recognition, and recommender systems.

In modern data analysis, we occasionally face a number of interesting challenges: our data may come from multiple heterogeneous sources, our data may be spatiotemporal, have few samples but lie in high dimensions, or our data may have a huge number of samples and require a method to understand the complex model. These challenges require us to consider new developments in statistics and kernel methods. In this work, we show how kernel methods can help us solve these challenges where our proposed solutions also involve other topics including domain adaptation, regularized statistics, deep learning, and deep Gaussian processes.

In Chapter 2, we study the problem of analyzing data from multiple heterogeneous sources. We derive a framework with a graphical causal model and maximum mean discrepancy to eliminate the biases between different datasets and to combine multiple datasets together in a systematic way for increased sample size and for improved statistical power. We use the framework for a problem motivated by Alzheimer's Disease research and show that we can successfully combine two datasets from different research centers to derive more accurate and consistent results.

In Chapter 3, we continue our study on the problem of the analysis of multiple

heterogeneous datasets but with a different focus. We derive new hypothesis tests and theoretical analysis to understand when it is beneficial to combine multiple datasets together, both in the low dimension setting and high dimension setting. We find that the problem is a bias-variance trade-off. When the reduction on the variance, which may due to increased sample size, is more than the increase on the bias from heterogeneous datasets, it is beneficial to combine different datasets even though they come from different sources.

In Chapter 4, we study how to build a nonparametric model for spatiotemporal data when the samples are few but the dimension is high. We apply the framework on a Chicago crime dataset to understand the occurrence of crimes and it's transmission between various Chicago communities as time goes. The multiple communities and nonparametric kernel class make our model lie in high dimension but the limited crime events make our sample size small. We solve the problem by adding regularizations for the kernel-based models and we build a solid theoretical foundation to understand the behavior of our models for high-dimensional spatiotemporal data.

Finally, in Chapter 5, we propose a framework to understand the flow of the information in deep probabilistic models, which includes Bayesian neural networks, deep Gaussian processes, deep kernel learning, and others. On the one hand, we show that we can understand the information flow in deep probabilistic models using kernels and statistical structure assumptions and we show the relation between Bayesian neural networks and deep Gaussian processes through theoretical analysis. On the other hand, our framework points out a way to extend additive models, hierarchical models and other statistical models with structure assumptions to the deep structure so that we can use modern high computation power like deep learning while maintaining the good properties of statistical models including interpretability and uncertainty estimate.

1.1 Kernel methods

In statistics and machine learning, linear models are used to capture the relations and patterns in the data. When the relation becomes complicated, linear models are often not rich or expressive enough and we turn to kernel methods. In smoothing spline models, a kernel is used to define a non-parametric class of the function. Given samples with predictors $\{\mathbf{x}_i\}_{i=1}^n$ and responses $\{y_i\}_{i=1}^n$, a linear model assumes that

$$y = \beta\mathbf{x} + \epsilon,$$

where β are coefficients and $\epsilon \sim N(0, \sigma^2)$. Meanwhile, a smoothing spline model Wahba (1990) considers that

$$y = f(\mathbf{x}) + \epsilon,$$

where $f(\mathbf{x})$ belongs to a function class determined by a kernel $\mathcal{K}$ and can be represented by $f(\mathbf{x}) = \sum_{i=1}^n \alpha_i \mathcal{K}(\mathbf{x}, \mathbf{x}_i)$, where α are coefficients. It helps us to capture the non-linear relation between predictors and responses. Through smoothing spline models, we can fit a smoothing curve or surface to understand the relation between the response and predictors.

In Gaussian process Rasmussen (2003), we consider that the joint probability of $\{(f(\mathbf{x}_i), y_i)\}_{i=1}^n$ is

$$p(\{(f(\mathbf{x}_i), y_i)\}_{i=1}^n) = (\prod_{i=1}^n p(y_i|f(\mathbf{x}_i)))p(\{f(\mathbf{x}_i)\}_{i=1}^n),$$

where $p(y_i|f(\mathbf{x}_i)) \sim N(f(\mathbf{x}_i), \sigma^2)$ and $p(f(\mathbf{x}_1), \cdots, f(\mathbf{x}_n)) \sim N(0, \mathcal{K}(\{\mathbf{x}_{i=1}^n\}, \{\mathbf{x}_{i=1}^n\}))$. The kernel is used to define the covariance matrix for the Gaussian processes so that one can understand the nonlinear trend of the data in a Bayesian manner.

In support vector machine Cortes and Vapnik (1995), a kernel is used to define a non-linear map to transform the predictors $\mathbf{x}$ into a high-dimensional space as

$\phi(\mathbf{x})$, where the kernel is the inner product of the transformed features $\mathcal{K}(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^{\mathsf{T}}\phi(\mathbf{x}')$. Through using kernels in support vector machine, one can find a hyperplane in the high dimensional space to distinguish data from different classes. All these models have been successful over the past two decades in applications for biomedical science, finance, pattern recognition, and recommender systems.

In modern data analysis, we face a number of interesting challenges: our data may come from multiple heterogeneous sources, our data may have few samples but lie in high dimension and is spatiotemporal, and our data may have a huge number of samples and require a method to understand the complex model. These challenges require us to consider new developments in statistics and kernel methods. In this work, we show how kernel methods can help us solve these challenges where our proposed solutions also involve other topics including domain adaptation, regularized statistics, deep learning, and deep Gaussian processes.

1.2 Data is from multiple heterogeneous sources

In Chapter 2 Zhou et al. (2016, 2018a), we study the problem of analyzing data from multiple heterogeneous sources. Many studies that involve human subjects are constrained by the number of samples that can be obtained when the disease population of interest is small, when the measurement of interest is difficult to obtain or when other logistic or financial constraints are present that prohibit large scale studies Fortin and Currie (2013); Buerger et al. (2009). For example, in Alzheimer's Disease (AD) research, cerebrospinal fluid (CSF) measurements from the lumbar puncture (LP) may be limited by participant willingness to undergo LP and institutional capability to routinely perform the procedure in a research setting. These issues are not restricted to biomedical studies, and variously manifest in machine learning and computer vision where distinct datasets must be pooled, e.g., for training a statistical model. To solve these issues, one possible solution is to identify and pool several similar datasets across multiple sites Carrillo et al. (2013b), and the larger sample sizes of the pooled dataset will enable investigating potentially interesting scientific questions that may not otherwise be possible with smaller

single-site cohorts. However, the datasets from multiple heterogeneous sources usually have different distributions and their combination may mask the signal. For example, the assays for proteins amyloid beta 1-42 and tau (the hallmark features of AD pathology) are known to vary widely between assay product type, and within a specific type of assay from differences in batch composition Vanderstichele et al. (2012). That motivates us to derive a kernel-based framework to analyze data from multiple heterogeneous sources in Chapter 2 Zhou et al. (2016, 2018a), that can account and eliminate the biases between various datasets.

We find that kernel methods can be very useful in solving this problem. The framework is built on a popular non-parametric quantity, maximum mean discrepancy (MMD), which is the mean difference between two datasets in the kernel space. The kernel space helps us to find a transformation that we can use to align two datasets from different sources and eliminate the biases between them. The framework also uses the graphical causal model to deal with the practical situation where the covariates for different sites (or studies) are not exactly the same (e.g., age range of cohorts may vary). The work also includes consistency properties, an identifiability condition and a hypothesis test to check model accuracy. In summary, we derive a framework with maximum mean discrepancy and graphical causal model to eliminate the biases between different datasets and to combine multiple datasets together in a systematic way for increased sample size and for improved statistical power. We use the framework on Alzheimer's Disease research and show that we successfully combine two datasets from different research centers to derive more accurate and consistent results.

In Chapter 3 Zhou et al. (2017), we continue our analysis of multiple heterogeneous datasets but with a different focus. We derive new hypothesis tests and theoretical analysis to answer when it is beneficial to combine multiple datasets together, both in the low dimension setting and high dimension setting. We find that the problem is a bias-variance trade-off. When the reduction on the variance, which may due to increased sample size, is more than the increase on the bias from heterogeneous datasets, it is beneficial to combine different datasets even though they come from different sources.

1.3 Data is spatiotemporal and high-dimensional

In Chapter 4 Zhou and Raskutti (2019), we study how to build a nonparametric model for spatiotemporal data when the samples are few but the dimension is high. Multi-variate time series data (spatiotemporal data) arise in a number of settings including neuroscience (Ding et al. (2011)), finance (Rydberg and Shephard (1999)), social networks (Zhou et al. (2013)) and others. A fundamental question associated with multi-variate time series data is to quantify influence between different players or nodes in the network (e.g. how do firing events in one region of the brain trigger another, how does a change in stock price for one company influence others). To address such a question requires *estimation of an influence network* between the d players or nodes. Two challenges that arise in estimating such an influence network are (i) developing a suitable network model; and (ii) providing theoretical guarantees for estimating such a network model when the number of nodes d is large.

Instead of using parametric approaches, we consider a non-parametric sparse additive model Raskutti et al. (2012); Ravikumar et al. (2010) for spatiotemporal data, which can capture non-linear effects such as saturation. In Chapter 4 Zhou and Raskutti (2019), we consider samples generated from a *non-parametric sparse additive auto-regressive model*, generated by the generalized linear model (GLM),

$$X_{t+1,j}|X_t \sim p\left(\nu_j + \sum_{k=1}^{d} f^*_{j,k}(X_{t,k})\right), \tag{1.1}$$

where $f^*_{j,k}$ is an unknown function belonging to a reproducing kernel Hilbert space $\mathcal{H}_{j,k}$, $p(.)$ is an exponential family probability distribution including the Gaussian, Poisson, Bernoulli and others to handle different data types.

In summary, we find kernel methods are very useful in analyzing spatiotemporal data. We provide a scalable non-parametric framework using technologies in sparse additive models for high-dimensional time series models that capture non-linear, non-parametric framework. Prior theoretical guarantees for sparse additive models have focused on the setting where samples are independent. In this work, we

analyze the convex penalized sparse and smooth estimator developed and analyzed in Koltchinskii and Yuan (2010); Raskutti et al. (2012) under the dependent Markov chain model (4.1). We demonstrate the flexibility and potential benefit of using the non-parametric approach through both a simulation study and real data example of Chicago crime dataset.

1.4 Data is huge and complex

Finally, in Chapter 5 Zhou et al. (2018b), we propose a framework to understand the flow of the information in deep probabilistic models including Bayesian neural networks, deep Gaussian processes, deep kernel learning, and others. We can collect a huge dataset and successfully use deep neural networks (DNNs) to identify patterns in the data over the past decades for applications including pattern recognition, documents translations, and recommender system. However, DNNs lack interpretability and probabilistic explanation. That leads to problems in applications including self-driving cars, scientific analysis, and finance. For example, when a car detects a person wrongly as a dog, it is important for us to understand why the DNN model fails and fix it. When a patient is detected with a disease, the doctor needs to tell the patient about the reason and how confident the result is. When a DNN model is used to predict a stock change, the reason and confidence for the prediction are very important. Should we buy a stock that may increase $1 with 90% chance or a stock that may increase $100 with 5% chance? All these issues require us to come up with a framework that can explain DNNs and provide uncertainties.

We find that kernel methods are very useful in solving this problem. Though the probabilistic formulation of DNNs is an existing area of research, called deep probabilistic models, it is unclear about how we can interpret these models and understand the flow of information in them. In our work, we derive a framework to understand the flow of information in deep probabilistic models and we show that the kernel methods play an important role. Our framework also points out a way to extend additive models, hierarchical models and other statistical models

with structure assumptions to deep structure so that we can use modern high computation power like deep learning while maintaining the good properties of statistical models including interpretability and uncertainty estimate.

2 MAXIMUM MEAN DISCREPANCY AND GRAPHICAL CAUSAL MODEL TO POOL AND ANALYZE DATASETS FROM MULTIPLE HETEROGENEOUS SOURCES

2.1 Introduction

In this Chapter, we discuss how to pool and analyze datasets from multiple heterogeneous sources. Many studies that involve human subjects are constrained by the number of samples that can be obtained when the disease population of interest is small, when the measurement of interest is difficult to obtain or when other logistic or financial constraints are present that prohibit large scale studies Fortin and Currie (2013); Buerger et al. (2009). For example, in Alzheimer's Disease (AD) research, cerebrospinal fluid (CSF) measurements from lumbar puncture (LP) may be limited by participant willingness to undergo LP and institutional capability to routinely perform the procedure in a research setting. The assays for amyloid beta 1-42 and tau (the hallmark features of AD pathology) are known to vary widely between assay product type, and within a specific type of assay from differences in batch composition Vanderstichele et al. (2012). Similarly, the expense of imaging exams may prohibit large scale investigations. While the sample sizes may be sufficient to evaluate the primary hypotheses, researchers may want to investigate secondary analyses focused on identifying subtle associations between specific predictors and the response variable Dubois et al. (2010); Vanderstichele et al. (2012). Such secondary analyses may be underpowered for the given sample sizes. One possible solution is to identify and pool several similar datasets across multiple sites Carrillo et al. (2013b). One hopes that the larger sample sizes of the pooled dataset will enable investigating potentially interesting scientific questions that may not otherwise be possible with smaller single site cohorts.

In practice, we find that direct pooling of already collected datasets in a post-hoc manner across multiple sites can be problematic due to differences in the distributions of one or more measures (or features) Verwey et al. (2009). In fact, even

when data acquisition is harmonized across sites, we may still need to deal with site-specific or method-specific effects on the measurements, such as the above noted example with CSF Mattsson et al. (2011), before the analysis can proceed Klunk et al. (2015); Carrillo et al. (2013a). For example, as discussed above, in AD studies, cerebrospinal fluid (CSF) measurements Wang et al. (2012) may not be easily pooled in the absence of gold standard reference materials that are common across assays (or sites) Vanderstichele et al. (2012). Such issues also arise in combining cognitive measures or transferring analysis results or models from one potentially large sized dataset to another. For example, cohort studies may administer different cognitive tests that assess the same underlying cognitive domain; therefore, thresholds used to categorize individuals into different disease status groups may not be easily transferred from one site to the other Carrillo et al. (2013b); Shaw et al. (2009). These issues are not restricted to biomedical studies, and variously manifest in machine learning and computer vision where distinct datasets must be pooled, e.g., for training a statistical model. While the literature on addressing sample selection bias and compensating for population characteristics differences is sizable Huang et al. (2007a); Bareinboim and Pearl (2016), statistical frameworks for resolving distributional shift to facilitate pooled analysis, essential in various applications, is less developed in comparison.

Deriving scientific conclusions from a unified analysis spanning multiple individual datasets is often accomplished in practice via so-called meta analysis approaches. Such an approach carefully collects research analyses/findings separately performed on the datasets and then aggregates individual analysis results through statistical models to come up with a final estimate of the parameters Lipsey and Wilson (2001). However, various assumptions in meta-analysis schemes may not always hold in practice and simple violations can lead to inaccurate scientific conclusions Greco et al. (2013); Stegenga (2011). Alternatively, if access to the actual data from individual studies is available, some pre-processing to harmonize the data followed by statistical analysis of the *pooled data* may be preferable in many cases. The pre-processing often utilizes methods that compensate (or correct) for distributional shift, to the extent possible. For example, ideas related to domain

shift in Baktashmotlagh et al. (2013); Ganin et al. (2016) and other results describe sophisticated models to improve prediction accuracy by correcting domain shift. What is less developed is a formal treatment explaining how confident we are that the shift across datasets has been successfully corrected (and consequently, the analysis can safely proceed), whether or not the correction can be improved if we were able to acquire more samples, what mathematical assumptions are needed, and whether the residual (say, after a correction step) is due to fewer than necessary samples or other violations of the underlying assumptions. The primary goal of this Chapter is to offer a formal treatment of these problems and derive the theoretical basis that can guide practical deployments.

In this Chapter, we present an in-depth theoretical study of distributional shift correction across datasets. That includes consistency properties, an identifiability condition and a hypothesis test to check model accuracy, using a discrepancy measure popular in the domain adaptation literature Baktashmotlagh et al. (2013); Ganin et al. (2016). We also provide an analysis based on a sub-sampling procedure, showing how these ideas can be modified to deal with the practical situation where the covariates for different sites (or studies) are not exactly the same (e.g., age range of cohorts may vary) — towards facilitating rigorous analysis of pooled datasets.

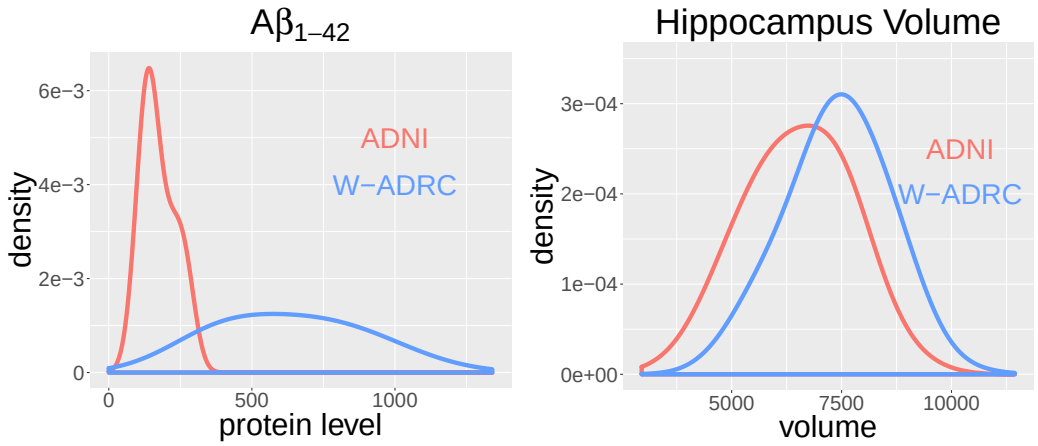
Our contributions.

i) give a precise condition to evaluate whether a distributional shift correction is identifiable;

ii) derive a subsampling procedure to separate distributional shift from other sources of variations such as sample selection bias and population characteristics differences;

iii) propose an algorithm based on a non-parametric quantity, Maximum Mean Discrepancy (MMD);

iv) present experiments showing how these ideas can facilitate Alzheimer's Disease (AD) biomarker research.



(a) (b)

Figure 2.1: (a) shows the distributional shift of $A\beta_{1\text{-}42}$ across ADNI and W-ADRC. (b) shows the distributional shift of hippocampus volume across ADNI and W-ADRC.

2.2 Problem Setting

Let us assume that we have data from two sites S and T, and the site-wise data corresponds to p different features. For presentation purposes, we will assume that the features include 8 CSF protein levels, denoted as X, acquired from each participant via a lumbar puncture. Since the absolute values of CSF measurements vary as a function of the assay instrumentation, we are interested in correcting the distributional shift to facilitate the analysis of the pooled dataset. But notice that there are at least two other factors that can influence the correction. First, S and T may have participants with age distributions that are not identical. It is known that age influences protein level measurements and therefore will affect our distributional shift correction. We denote the population characteristics which cause differences in age distributions as E_P (also called "transportability" in Bareinboim and Pearl (2016)). Similarly, while site S may include an almost equal split of individuals with and without disease, healthy individuals may be overrepresented in site T. We denote this bias in sample selection between two datasets as E_B, which also influences X Bareinboim and Pearl (2016). Therefore, the actual distributions

of observed CSF protein levels in the two datasets, X_S and X_T are $P(X_S|E_P, E_B)$ and $P(X_T|E_P, E_B)$, respectively. If we only have access to X_S and X_T but no other variables related to E_P and E_B, then correcting the distributional shift between X_S and X_T is difficult. However, the problem is identifiable when we have age and diagnosis status relevant for the variables E_P and E_B. In fact, we can specify the condition when the correction is identifiable. We briefly review some concepts related to graphical causal model and d-separation rules, and then state the identifiability condition.

Graphical Causal Model

A graphical causal model is represented by a directed acyclic graph (DAG), which consists of three types of entities: variables (nodes), arrows (edges), and missing arrows. DAGs are useful visual representations of a domain expert's assumptions regarding causal relationships explaining the data generation process Elwert (2013). In Fig. 2.2(a), we show an example. Arrows in the graph represent possible direct causal effects between pairs of variables. For example, the arrow from I to O_1 means that I exerts a direct causal influence on O_1. The absence of an arrow represents an assumption of no direct causal effect between the two variables Elwert (2013). The missing arrow from I to J denotes the absence of a direct causal effect of I on J. Fig. 2.2(b) shows an example for our data analysis task where the DAGs depict causal relations between age, sex, CSF, diagnosis status and other variables. Here, age, sex and other endogenous variables influence the CSF measurements X, which influences the diagnosis status D. The population characteristic difference E_P only has a direct causal effect on age, whereas the sample selection bias E_B is only directly related to diagnosis status D for each specific study or site. Note that a graphical causal model is nonparametric and makes no other assumptions about the distribution of variables, the functional form of direct effects or the magnitude of causal effects.

Next, we introduce a useful concept called d-separation Pearl (2014) using the model in Fig. 2.2(a) as an example. If two variables I and J are d-separated by a

set of variables Z, then they are conditionally independent, given Z. A path is a sequential set of connected nodes, independent of the directionality of the arrows. A "collider" on a path is a node with two arrows along the path pointing into it (see $O_5 \to O_3 \leftarrow O_6$ in Fig. 2.2(a)). Otherwise, the node is a noncollider on the path.

Definition 2.1 (d-separation (Pearl (2014)))**.** *A path* p *between two variables,* I *and* J*, is said to be blocked by a set of variables* Z *if either:* ***(1)*** p *contains a noncollider that is in* Z *or* ***(2)*** p *contains a collider node that is outside* Z *and has no descendant in* Z*. We say that* I *and* J *are d-separated by* Z *if any path between them is "blocked" by* Z*.*

For example, in Fig. 2.2(a), I and J are d-separated by $Z = \{O_1, O_2, O_3\}$. First, after including $\{O_1, O_3\}$ in Z, all paths are blocked due to rule (1) except the path $p_1 : I \leftarrow O_2 \leftarrow O_5 \to O_3 \leftarrow O_6 \to O_4 \to J$. The path p_1 stays unblocked because (1) no noncollider on that path is in Z and (2) the only collider O_3 on p_1 is in Z. So, we can include one of $\{O_2, O_5, O_6, O_4\}$ on the path into Z to "block" it.

2.3 Identifiability Condition

We can now present a condition describing when distributional shift correction across sites is identifiable, even with the concurrent influence of sample selection bias and population characteristic differences on the measurements X.

Theorem 2.2. *The distribution shift correction is identifiable if there exists a known set of variables* Z *such that the following three conditions are all concurrently satisfied:*

1) Z *d-separates* X *and* E_B *(sample selection bias) and also d-separates* X *and* E_P *(population characteristic difference);*

2) *The conditional probability* $\mathbb{P}(X|Z)$*, after appropriate transformations on* X*, is the same across multiple participating sites (*S *and* T*);*

3) *The distribution of* Z *has a non-trivial overlap across multiple sites (*S *and* T*), which means that there exists an interval* $[a, b]$ *such that* $\mathbb{P}(a \leqslant Z \leqslant b) \geqslant 0.5$ *for all sites.*

From Fig. 2.2(b) and Tab. 2.1, we can check that $Z = \{D, age\}$ satisfies Thm. 2.2. Condition (1) is satisfied by noticing that Z d-separates X and the nodes

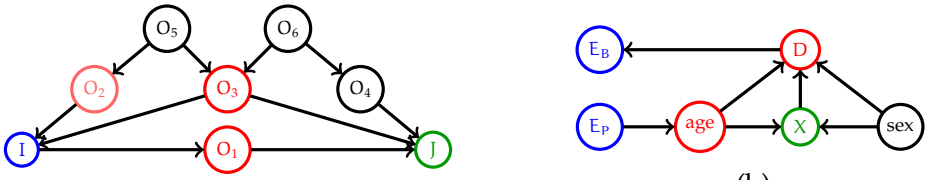


Figure 2.2: Plot (a) is an example of a graphical causal model. The colored nodes are an example of a d-separation rule where I and J are d-separated by $\{O_1, O_2, O_3\}$. Plot (b) is the graphical causal model for our CSF data analysis example. Here, the population characteristics difference E_P only has a direct causal effect on the age distribution. The sample selection bias E_B is only directly related to diagnosis status D for each specific study. Nodes denoting age, sex influence the CSF measurements denoted by X, which then influence the diagnosis status D. The CSF measurements X and the nodes E_P and E_B are d-separated by diagnosis status D and age.

E_P and E_B. If all sites collect samples similarly, $\mathbb{P}(X|Z)$ will be the same (e.g., $\mathbb{P}(X|D = AD, age = 80)$). From Fig. 2.2(b), variations denoted by E_P and E_B only influence the marginal distributions of D and age but have no effect on the causal relation/function among variables, e.g., $\mathbb{P}(X|Z)$. The distributional shift of X can be corrected after some transformation, therefore, condition (2) holds. Finally, we will see (e.g., Tab. 2.1) that the disease status and age distributions have a non-trivial overlap across the two datasets, therefore, condition (3) also holds.

In practice, it is useful to seek a d-separating set of variables Z with the fewest variables such that we can sacrifice (or leave out) the fewest samples to separate distributional shift from the other variations E_P and E_B. Finding a minimal d-separating set can be solved as a maximum flow problem Acid and De Campos (1996). In practice, if the causal model is not too complicated, one may even find a d-separating set Z manually. Then, it can be transformed into the problem of "blocking" two nodes in an undirected graph with the fewest blocks Tian et al. (1998).

The algorithm to find a minimal d-separating set. Before stating the algorithm, we introduce some notations first. For any node set A, we define $\mathcal{A}_n(A)$ to be the

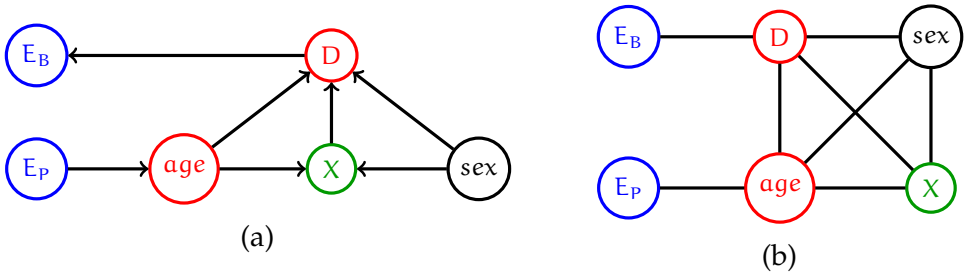


Figure 2.3: In this graphical diagram, (a) represents the graphical causal model for our AD study example. CSF X and other variations E_P, E_B are d-separated by diagnosis information D and age age, whereas (b) represents the moral graph of the subgraph on X, E_P, E_B and their ancestors, that is, $m\mathcal{DAG}_{\mathcal{A}_n(X\cup E_P\cup E_B)}$.

ancestral set containing set A, that is, $\mathcal{A}_n(A) = A \cup (\cup_{u\in A}\{\text{all ancestors of } u\})$. We call the directed subgraph composed only of nodes from $\mathcal{A}_n(A)$ as $\mathcal{DAG}_{\mathcal{A}_n(A)}$. A so-called moral graph is formed from a directed acyclic graph by adding edges between all pairs of nodes that have a common child, and then making all edges in the graph undirected. We denote the moral graph of $\mathcal{DAG}_{\mathcal{A}_n(A)}$ as $m\mathcal{DAG}_{\mathcal{A}_n(A)}$.

In order to find a minimal d-separating set Z for the measurements of interest X and bias nodes E_P and E_B, we only need to consider this question on the undirected graph $(\mathcal{DAG}_{\mathcal{A}_n(X\cup E_P\cup E_B)})$ instead of the full directed acyclic graph (as shown in Tian et al. (1998)). We give an example for finding the minimal d-separating set of the graph, which is shown in Fig. 2.3. Instead of using Fig. 2.3, the minimal d-separating set can be found from Fig. 2.3(b), which represents $m\mathcal{DAG}_{\mathcal{A}_n(X\cup E_P\cup E_B)}$.

The authors Tian et al. (1998) show that searching a d-separating set for X and the bias nodes on the original DAG is equivalent to finding a node set that can block any path between X and bias nodes in the moral graph of the DAG. Therefore, we can run a Breadth First Search (BFS) algorithm on the undirected graph $m\mathcal{DAG}_{\mathcal{A}_n(X\cup E_P\cup E_B)}$ to check whether Z is the minimal d-separating set and adjust it if it is not. For our example in Fig. 2.3, by using BFS we can see that $Z = \{D, age\}$ will block any path from X to the bias nodes E_P and E_B in Fig. 2.3(b).

2.4 Tests for Correcting Distributional Shift

We now describe an algorithm to correct distributional shift if it is identifiable (Thm. 2.2). We start our discussion by first assuming that the two to-be-pooled datasets, S and T, only include a distributional shift in the features (e.g., due to measurement or site-specific nuisance factors) and involve no other sampling biases or confounds (i.e., E_P and E_B). Later, we present a subsampling framework to extend the algorithm to the case when other variations co-occur and also contribute to the shift. We calculate the distributional shift correction by identifying a parametric transformation on the site-wise samples from S and T. We assume that site S provides n_S samples $X_S = (x_S^1, x_S^2, ..., x_S^{n_S})$, given by a distribution P_S and T provides n_T samples X_T with a distribution P_T.

Let us denote the transformation on X_S as $h^\lambda(\cdot)$ and the transformation on X_T as $g^\theta(\cdot)$, characterized by the unknown parameters λ and θ respectively. For example, if we choose $h^\lambda(\cdot)$ to be an affine transformation with parameters $\lambda := W$, it maps any value x to Wx; that is $h^W(\cdot) : x \to Wx$. The algorithm seeks to find a pair of transformations such that distributions of two datasets are matched (corrected) after the transformations are applied. We use maximum mean discrepancy (MMD) as a measure of difference between the two (transformed) distributions. The MMD is expressed as a function of two distributions P_S, P_T as

$$\mathcal{MMD}(P_S, P_T) = \|\mathbb{E}_{X\sim P_S}\mathcal{K}(X, \cdot) - \mathbb{E}_{X\sim P_T}\mathcal{K}(X, \cdot)\|_{\mathcal{H}}$$

which is defined using a Reproducing Kernel Hilbert Space (RKHS) with norm $\|\cdot\|_{\mathcal{H}}$ and kernel $\mathcal{K}$. MMD can also be considered as the mean difference between two distributions after kernel embedding, and has several desirable properties, for example, it is zero if and only if two distributions are identical Gretton et al. (2012). One requirement, however, is that the kernel has to be characteristic and specific choices may be guided by the application Gretton et al. (2012). The empirical version

of MMD can be calculated with samples X_S, X_T as

$$\widehat{\mathcal{MMD}}(X_S, X_T) = \|\frac{1}{n_S}\sum_{i=1}^{n_S}\mathcal{K}(x_S^i, \cdot) - \frac{1}{n_T}\sum_{j=1}^{n_T}\mathcal{K}(x_T^j, \cdot)\|_{\mathcal{H}}$$

Recall that our algorithm is trying to match the two distributions *after* applying the parametric transformations $h^\lambda(\cdot)$ and $g^\theta(\cdot)$. Therefore, we estimate parameters λ and θ using the empirical MMD by searching for a minimum value, e.g., using stochastic gradient descent,

$$(\hat{\lambda}, \hat{\theta}) = \arg\min_{\lambda\in\Omega_\lambda, \theta\in\Omega_\theta} \widehat{\mathcal{MMD}}(h^\lambda(X_S), g^\theta(X_T)) \tag{2.1}$$

The class of transformations we will choose for a specific application should be informed by domain knowledge, but in general, simpler transformation classes are preferable.

Optimization

We give a special case of the optimization problem when we set $\mathcal{K}$ to be a Gaussian kernel, $g^\beta(.)$ as the identity and $h^W(x) = WT_r(x)$ as a linear function for parameters W. Here, $T_r(x)$ is a known transformation on x, for example, $T_r(x) = x^2$ relates to the second order polynomial transformation. Recall that the general form of the optimization is

$$(\hat{\lambda}, \hat{\theta}) = \arg\min_{\lambda\in\Omega_\lambda, \theta\in\Omega_\theta} \|\frac{1}{n_S}\sum_{i=1}^{n_S}\mathcal{K}(h(x_S^i)^\lambda, \cdot) - \frac{1}{n_T}\sum_{j=1}^{n_T}\mathcal{K}(g(x_T^j)^\theta, \cdot)\|_{\mathcal{H}}^2 \tag{2.2}$$

For our example case, the simplified objective can be written as

$$\begin{aligned}
\hat{W} &= \arg\min_{W\in\Omega_W} F(W) \\
&= \arg\min_{W\in\Omega_W} \|\frac{1}{n_S}\sum_{i=1}^{n_S}\mathcal{K}(WT_r(x_S^i),\cdot) - \frac{1}{n_T}\sum_{j=1}^{n_T}\mathcal{K}(x_T^j,\cdot)\|_{\mathcal{H}}^2 \\
&= \arg\min_{W\in\Omega_W} \frac{1}{n_S^2}\sum_{i=1}^{n_S}\sum_{j=1}^{n_S}\mathcal{K}(WT_r(x_S^i),WT_r(x_S^j)) + \frac{1}{n_T^2}\sum_{i=1}^{n_T}\sum_{j=1}^{n_T}\mathcal{K}(x_T^i,x_T^j) \\
&\quad - \frac{2}{n_S n_T}\sum_{i=1}^{n_S}\sum_{j=1}^{n_T}\mathcal{K}(WT_r(x_S^i),x_T^j) \\
&= \arg\min_{W\in\Omega_W} \frac{1}{n_S^2}\sum_{i=1}^{n_S}\sum_{j=1}^{n_S}\exp(\|W(T_r(x_S^i)-T_r(x_S^j))\|_2^2) \\
&\quad - \frac{2}{n_S n_T}\sum_{i=1}^{n_S}\sum_{j=1}^{n_T}\exp(\|WT_r(x_S^i)-x_T^j\|_2^2)
\end{aligned} \tag{2.3}$$

This is a continuous optimization and its gradient with respect to F is:

$$\begin{aligned}
\nabla F(W) =& \frac{2}{n_S^2}\sum_{i=1}^{n_S}\sum_{j=1}^{n_S}\exp(\|W(T_r(x_S^i)-T_r(x_S^j))\|_2^2)tr(W^TW,(T_r(x_S^i)-T_r(x_S^j))(T_r(x_S^i)-T_r(x_S^j))^T) \\
&- \frac{4}{n_S n_T}\sum_{i=1}^{n_S}\sum_{j=1}^{n_T}\exp(\|WT_r(x_S^i)-x_T^j\|_2^2)(T_r(x_S^i)^TW^T)(WT_r(x_S^i)-x_T^j)
\end{aligned} \tag{2.4}$$

Since the objective is continuous, the optimization can be performed by gradient descent or stochastic gradient descent. A useful prior based on some domain knowledge can also help the optimization scheme since the objective is non-convex.

We now provide a result to setup a confidence region to choose a good initial point for the optimization.

Lemma 2.3. *Under* H_0, *the identity* $g^\theta(.)$ *with* $h^W(x) = WT_r(x)$, *we have*

$$\Omega_W := \left\{ W | \frac{1}{\min(n_S,n_T)} \sum_{i=1}^{\min(n_S,n_T)} \|x_T^i - WT_r(x_S^i)\|_2^2 \leqslant 3tr(\mathbb{VAR}(P_T)) + \epsilon \right\},$$

where $\mathbb{VAR}$ *is the variance and* $tr(\cdot)$ *is the trace. For any* $\epsilon, \alpha > 0$ *and a sufficiently large sample size, a neighborhood of the true* W_0 *is contained in* Ω_W *with probability at least* $1 - \alpha$.

Hypothesis tests and consistency

We now show that the estimators $\hat{\lambda}$ and $\hat{\theta}$ are consistent. First, we describe some assumptions. We define the search region for λ as Ω_λ which belongs to $\mathbb{R}^{p_\lambda}$; similarly, the search region for θ is defined as $\Omega_\theta \in \mathbb{R}^{p_\theta}$. In our results, we assume that $\Omega_\lambda, \Omega_\theta$ live in Euclidean space. The results, however, may be generalized to other spaces with similar techniques.

Assumptions 2.4. *We require three assumptions for the consistency results.*
(1) The search regions $\Omega_\lambda, \Omega_\theta$ *of* λ, θ *are bounded.*
(2) The kernel $\mathcal{K}$ *of RKHS used for MMD is non-negative, characteristic and bounded by a constant* k.
(3a) $\|\mathcal{K}(h^{\lambda_1}(x), \cdot) - \mathcal{K}(h^{\lambda_2}(x), \cdot)\|_H \leqslant L_h \|\lambda_1 - \lambda_2\|_2^{r_h}$ *with constants* L_h, r_h, *for any* x *in the support of* P_S *and* $\forall \lambda_1, \lambda_2 \in \Omega_\lambda$
(3b) $\|\mathcal{K}(g^{\theta_1}(x), \cdot) - \mathcal{K}(g^{\theta_2}(x), \cdot)\|_H \leqslant L_g \|\theta_1 - \theta_2\|^{r_g}$ *with constants* L_g, r_g, *for any* x *in the support of* P_T *and* $\forall \theta_1, \theta_2 \in \Omega_\theta$

The Assumption 2.4 (2) is used for most consistency analyses based on MMD Gretton et al. (2012). The Assumption 2.4 (3a) and (3b) are satisfied for a big class of transformations with differentiable radial basis kernel. We have following sufficient conditions to satisfy Assumptions 2.4 (3a) and (3b).

Lemma 2.5. *If the following two conditions are satisfied, then Assumptions 2.4 (3a) and (3b) hold.*
(1) The kernel function $\mathcal{K}(\|x - y\|_2)$ *is a radial basis kernel where* $\partial\mathcal{K}(.)$ *is bounded in a neighborhood of* 0.
(2) The transformation $h^\lambda(x)$ *is Holder-continuous as a function of* λ *with ratio* r_h *for any* x *in the support of* P_S. *Similarly, the transformation* $g^\theta(x)$ *is Holder-continuous as a function of* θ *with ratio* r_g *for any* x *in the support of* P_T.

We now show that the estimators $\hat{\lambda}$ and $\hat{\theta}$ are consistent.

Theorem 2.6. *Under mild assumptions 2.4, if there is a* λ_0, θ_0 *such that* $h^{\lambda_0}(X_S)$ *and* $g^{\theta_0}(X_T)$ *have the same distribution, then*

$$\mathcal{MMD}(h^{\hat{\lambda}}(X_S), g^{\hat{\theta}}(X_T)) \to 0$$

with the rate $\max(\frac{\sqrt{\log(n_S)}}{\sqrt{n_S}}, \frac{\sqrt{\log(n_T)}}{\sqrt{n_T}})$. *If* λ_0, θ_0 *are unique, then the estimators* $\hat{\lambda}, \hat{\theta}$ *are consistent.*

Remark. In various applications (including our experiments), we may choose one class of transformations $h^{\lambda}(x)$ to be the identity transformation and transform samples in the other dataset to match the reference dataset.

The foregoing discussion and the theorem assumes that the two distributions can be matched via *some* unknown transformation. This may not always be true and it is important, in practice, to identify when the datasets cannot be pooled, for the specified class of transformations. Next, we provide a hypothesis test to answer this question. Let us define

$$H_0 : \text{There exists } \lambda, \theta \text{ such that } h^{\lambda}(X_S) \text{ and } g^{\theta}(X_T) \text{ match}$$
$$H_A : \text{There is no } \lambda, \theta \text{ such that } h^{\lambda}(X_S) \text{ and } g^{\theta}(X_T) \text{ match}$$

The test statistics can be obtained by plugging $\hat{\lambda}$, $\hat{\theta}$ into the empirical MMD calculation as,

$$\widehat{\mathcal{MMD}}_{\text{best}} = \widehat{\mathcal{MMD}}(h^{\hat{\lambda}}(X_S), g^{\hat{\theta}}(X_T))$$

We can show that the hypothesis test is consistent.

Theorem 2.7. *Given Assumptions 2.4, when* H_0 *is true, with probability at least* $1 - \alpha$,

$$\widehat{\mathcal{MMD}}(h^{\hat{\lambda}}(X_S), g^{\hat{\theta}}(X_T)) \leqslant \sqrt{\frac{2k(n_S + n_T)\log \alpha^{-1}}{n_S n_T}} + 2\sqrt{\frac{k}{n_S}} + 2\sqrt{\frac{k}{n_T}}$$

When H_A *is true, with probability at least* $1-\alpha$,

$$|\widehat{\mathcal{MMD}}(h^{\hat{\lambda}}(X_S), g^{\hat{\theta}}(X_T)) - C_*| \leqslant \sqrt{\frac{k}{n_S}}\left(4+\sqrt{C^{h,\alpha}+\frac{d_\lambda}{2r_h}\log n_S}\right)+\sqrt{\frac{k}{n_T}}\left(4+\sqrt{C^{g,\alpha}+\frac{d_\theta}{2r_g}\log n_T}\right)$$

where $C_* = \min_{\lambda,\theta}\mathcal{MMD}(h^\lambda(P_S), g^\theta(P_T))$ *is a positive constant when* H_A *holds Gretton et al. (2012). Here,* $C^{h,\alpha} = \log(2|\Omega_\lambda|)+\log\alpha^{-1}+\frac{d_\lambda}{r_h}\log\frac{L_h}{\sqrt{k}}$ *and* $C^{g,\theta} = \log(2|\Omega_\theta|)+\log\alpha^{-1}+\frac{d_\theta}{r_g}\log\frac{L_g}{\sqrt{k}}$

The test can provide guidance on whether the distributional shift has been successfully corrected. If the test suggests the alternative hypothesis, one may consider adjusting the transformation class $h^\lambda(\cdot)$ and $g^\theta(\cdot)$, other factors such as sample selection bias and population attribute difference, or decide against pooling.

The threshold given in Thm. 2.7 can be used for performing hypothesis tests when the sample size is large enough. But when the sample sizes are small to moderate, a data-driven method may perform better. The same observation is discussed in related papers Gretton et al. (2012) for the MMD-based two sample test.

For the data-driven method, we require one transformation class to always be the identity. In other words, we consider $g^\theta(X_T)$ to always be X_T itself and we transform X_S using $h^{\hat{\lambda}}(X_S)$ to match X_T. Before presenting the method, let us recall our definition of the null and alternative hypotheses.

$$H_0 : \text{There exists a } \lambda \text{ such that } h^\lambda(X_S) \text{ matches } X_T$$
$$H_A : \text{There is no } \lambda \text{ which matches } h^\lambda(X_S) \text{ and } X_T$$

To test these hypotheses, we construct a bootstrap-type procedure here to calculate p-values:

Algorithm 1 Bootstrap-type hypothesis testing procedure

1: Define B to be the total number of bootstraps
2: **for** b in $1 : B$ **do**
3: Randomly generate n_S and n_T samples: $\tilde{X}_S^b, \tilde{X}_T^b$ from empirical distribution based on X_T with replacement.
4: Calculate the empirical distance between two distribution, mmd^b, by $\mathrm{mmd}^b = \widehat{\mathcal{MMD}}(\tilde{X}_S^b, \tilde{X}_T^b)$
5: **end for**
6: Compute p-value given the test statistics $\widehat{\mathcal{MMD}}(h^{\hat{\lambda}}(X_S), X_T)$ based on the empirical distributions of $(\mathrm{mmd}^1, ..., \mathrm{mmd}^B)$.

Remark. We can speed up the loop in Alg. 1 by computing distances between $n_S + n_T$ samples and permuting them for each iteration.
We now show that our procedure is a valid hypothesis test given the following results:

Lemma 2.8. *Given n_S samples $\tilde{X}_S$ from P_T, we have that (from Thm. 7 in Gretton et al. (2012)) with probability at least $1 - \alpha$,*

$$\widehat{\mathcal{MMD}}(\tilde{X}_S, X_T) \leqslant \sqrt{\frac{2k(n_S + n_T)\log \alpha^{-1}}{n_S n_T}} + 2\sqrt{\frac{k}{n_S}} + 2\sqrt{\frac{k}{n_T}}. \tag{2.5}$$

Meanwhile, if H_0 is true: there exists a λ_0 such that $h^{\lambda_0}(X_S)$ come from the same distribution as the underlying distribution P_T for X_T. Therefore, $\widehat{\mathcal{MMD}}(h^{\lambda_0}(X_S), X_T)$ is identically distributed as $\widehat{\mathcal{MMD}}(\tilde{X}_S, X_T)$,

$$\widehat{\mathcal{MMD}}(h^{\lambda_0}(X_S), X_T) \sim \widehat{\mathcal{MMD}}(\tilde{X}_S, X_T). \tag{2.6}$$

[2.5] tells us that the bootstrap empirical distribution we construct is bounded by some constants converging to 0 with probability at least $1 - \alpha$. Therefore, we can reject the null hypothesis when H_A holds because $\widehat{\mathcal{MMD}}(h^{\hat{\lambda}}(X_S), X_T)$ converges to a positive constant asymptotically with high probability from Thm. 2.7. Also, we can control the type I error from our bootstrap procedure in the following way. For

a critical value (related to a significance level α) calculated from the bootstrap empirical distribution, when H_0 is true, $\widehat{\mathcal{MMD}}(h^{\lambda_0}(X_S), X_T)$ is smaller than the critical value with probability at least $1-\alpha$ because of [2.6]. Further $\widehat{\mathcal{MMD}}(h^{\hat{\lambda}}(X_S), X_T)$ must be smaller than $\widehat{\mathcal{MMD}}(h^{\lambda_0}(X_S), X_T)$ since the former quantity is the minimum in the search region Ω_λ. Therefore, the type 1 error for our hypothesis test is well-controlled.

Next, we introduce a subsampling scheme to correct distributional shift when other contributors to the shift co-exist but the correction is still identifiable.

Subsampling Framework

When the test chooses H_A, one reason may be that one or more cohort-specific factors contribute in significant ways to the observed distributional shift between X_S and X_T. Recall that our earlier discussion suggests that the problem is identifiable if we can find a Z satisfying the conditions in Thm. 2.2. Then a subsampling procedure can potentially resolve the confound. The reason is that,

$$\mathbb{P}(X|E_P, E_B) = \mathbb{E}_{Z|E_P,E_B}[\mathbb{P}(X|Z, E_P, E_B)].$$

From Thm. 2.2, we know that $\mathbb{P}(X|Z, E_P, E_B) = \mathbb{P}(X|Z)$ which remains the same across sites after a suitable transformation. Therefore, simply by adjusting $\mathbb{P}(Z|E_P, E_B)$, the effects of the other factors on X can be controlled, except distributional shift. Such a sub-sampling scheme is widely used in addressing sample selection bias in other applications Gong et al. (2013) (also see Wang et al. (2017) on the sub-sampling scheme for reducing computational burden). In our setting, the motivation for using sub-sampling is similar, but it is used in the context of correcting distributional shift — *after* subsampling. Separately, since subsampling has been used in bagging to stabilize estimations and reduce variance (e.g., for random forests Wager et al. (2014)) we can directly obtain stable estimators and calculate their variance.

Specifics of subsampling. We divide X_S into d groups with sample sizes given as $(n_S^1, ..., n_S^d)$, i.e., $X_S = (x_S^{(1,1)}, ..., x_S^{(1,n_S^1)}, ..., x_S^{(d,1)}, ..., x_S^{(d,n_S^d)})$. Similarly, X_T is divided into groups with sample sizes given as $(n_T^1, ..., n_T^d)$, i.e., $X_T = (x_T^{(1,1)}, ..., x_T^{(1,n_T^1)},$

$..., x_T^{(d,1)}, ..., x_T^{(d,n_T^d)}$). The sub-sample sizes are $(s_1, s_2, ..., s_d)$ where $s_j \leqslant \min(n_S^j, n_T^j)$ for any $j = 1, ..., d$. Then, we generate subsamples for X_S and X_T and apply (2.1) sequentially. We run sub-sampling with replacement B times and denote each iteration's estimators as $\hat{\lambda}^b, \hat{\theta}^b$. Then, our final transformation estimators are given as $\hat{\lambda} = \frac{1}{B}\sum_{b=1}^{B}\hat{\lambda}^b$ and $\hat{\theta} = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}^b$.

Infinitesimal Jackknife Confidence Interval

In most scientific studies, we also want to obtain a confidence interval for the calculated transformations. In this case, however, there is no closed form solution and so we employ a bootstrap type method. Since subsampling already involves bootstrapping, using a simple bootstrap results in a product of bootstraps. Fortunately, a similar issue was encountered in bagging and an infinitesimal Jackknife method Efron (2014), was provided for random forests which works quite well Wager et al. (2014); Wager and Athey (2017). Inspired by this result, we use the infinitesimal Jackknife to estimate the variances of estimators $\hat{\lambda}$ and $\hat{\theta}$. The method cannot be directly applied here since it considers subsampling from one group whereas we need subsampling from multiple groups. We therefore extend the results to multiple groups.

Based on the subsampling scheme for X_S and X_T defined above, the multi-groups infinitesimal Jackknife estimators (IJ estimator) of variance is given as

Theorem 2.9. *Define* $g^b_{u(i,k)}$ *to be the number of appearances of* $x_u^{(i,k)}$ *in iteration* b. *Define* $\mathbb{COV}(g_{u(i,k)}, \lambda) = \frac{1}{B}\sum_{b=1}^{B}(\hat{\lambda}^b - \hat{\lambda})(g^b_{u(i,k)} - \frac{s_i}{n_u^i})$. *The IJ estimator of variance for* $\hat{\lambda}$ *is*

$$\mathbb{VAR}_{IJ}(\hat{\lambda}) = \sum_{u\in\{S,T\}}\sum_{i=1}^{d}\sum_{k=1}^{n_u^i}(\mathbb{COV}(g_{u(i,k)}, \lambda))^2$$

The procedure for $\hat{\theta}$ *is identical.*

Algorithm 2 Subsampling-MMD algorithm ($\mathcal{SSP}$)

1: Divide X_S and X_T separately into d groups by Z.
2: Decide subsample size $(s_1, s_2, ..., s_d)$.
3: **for** $b = 1$ to B **do**
4: Generate subsamples X_S^b from d groups of X_S.
5: Generate subsamples X_T^b from d groups of X_T.
6: $(\hat{\lambda}^b, \hat{\theta}^b) = \arg\min_{\lambda\in\Omega_\lambda, \theta\in\Omega_\theta} \widehat{\mathcal{MMD}}(h^\lambda(X_S^b), g^\theta(X_T^b))$
7: Calculate and record $g^b_{u(i,k)}$ for all u, i, k.
8: **end for**
9: Set $\hat{\lambda} = \frac{1}{B}\sum_{b=1}^{B} \hat{\lambda}^b$, $\hat{\theta} = \frac{1}{B}\sum_{b=1}^{B} \hat{\theta}^b$, calculate $\mathbb{VAR}_{IJ}(\hat{\lambda})$, $\mathbb{VAR}_{IJ}(\hat{\theta})$.

2.5 Experiments

In this section, we show the application of our framework for synthetic data and Alzheimer's disease study.

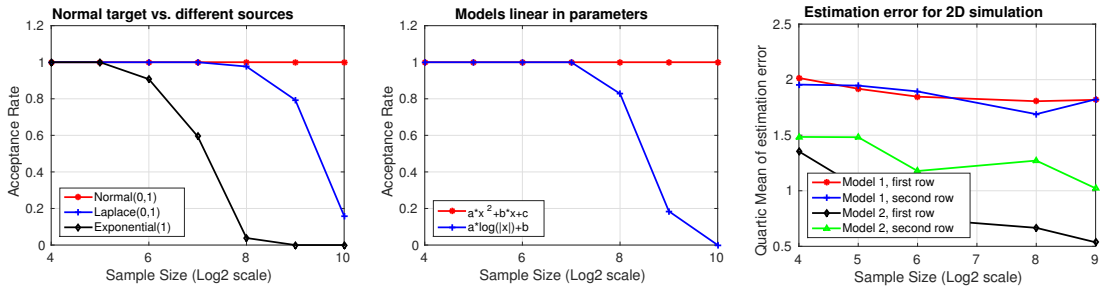
Synthetic data

The first set of experiments were designed to check the efficacy of the hypothesis test whereas the second experiment evaluated the estimation consistency.

Simulation for the Hypothesis Test

We generated samples from the standard normal distribution $N(0,1)$ to synthesize the first dataset X_S and use the normal distribution $N(10,2)$, with mean 10 and standard deviation 2, to synthesize another dataset X_T. Notice that, under the correct transformation class $h^{(a,b)}(X_S) = aX_S + b$, we can correct the distribution shift and we should accept H_0. We consider two types of variations that can potentially affect the correction and check whether the hypothesis test indeed rejects H_0 with high power.

1) In Fig. 2.4(a), we always choose the transformation class $h^{(a,b)}(X_S) = aX_S + b$ and one dataset X_T comprised of samples from $N(10,2)$. But we vary the generating distribution for the other dataset choosing between $N(0,1)$, $Laplace(0,1)$ and



(a) (b) (c)

Figure 2.4: (a,b) Acceptance rate for our hypothesis test when H_0 is true or false. (c) Error of our estimators under two different cases for generating distributions.

$\mathtt{Exponential}(1)$. We also vary the sample sizes. The hypothesis test with significance level 0.05 is performed on 100 repetitions and the acceptance rate curves are plotted. From Fig. 2.4(a) we see that our hypothesis test does accept H_0 at a high rate when it is true (red curve) and tends to reject H_0 with an increase in power as the sample size increases whenever the generating distributions are not Normal (blue and black curves).

2) In Fig. 2.4(b), we always choose one dataset X_T composed of samples from $N(10, 2)$ and the generating distribution for the other dataset X_S is set to be $N(0, 1)$. Then, we vary the transformation class $h^{(a,b,c)}(X_S)$ choosing between $aX_S^2 + bX_S + c$ and $a\log(|x|) + b$. Here, note that $aX_S^2 + bX_S + c$ includes the true transformation whereas $a\log(|x|) + b$ corresponds to an incorrect transformation class. Again, we vary the sample sizes and repeat the hypothesis test 100 times and plot the acceptance curves in Fig. 2.4(b). Similar to the first setting, we observe that our hypothesis test accepts H_0 at a high rate when it is true (red curve) and tends to reject H_0 with high power as the sample size increases in the setting where the transformation class is wrong (blue curve).

Simulation for Estimation Consistency

In this simulation, we assume that the distributional shift is the only variation across the datasets, X_S and X_T. We check the estimation consistency of the transformation

$h^{\hat{W}}(\cdot)$ which minimizes the MMD distance between $h^{\hat{W}}(X_S)$ and X_T.

In Fig. 2.4(c), we perform the experiments for two models. For model 1, the generating distributions for two datasets are,

$$X_S \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}\right)$$
$$X_T^{raw} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}\right)$$
$$X_T = \begin{pmatrix} 1 & 2 & 10 \\ 2 & 1 & -20 \end{pmatrix} \begin{pmatrix} X_T^{raw} \\ 1 \end{pmatrix}$$

In other words, we generate samples for X_S and X_T^{raw}, and we transform X_T^{raw} to get X_T. The transformation class we consider is

$$X_T = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \begin{pmatrix} X_S \\ 1 \end{pmatrix}$$

We define the quadratic mean of the estimation error for "Model 1, first row" in Fig. 2.4(c) as

$$\sqrt{\frac{(a_{11}-1)^2 + (a_{12}-2)^2 + (a_{13}-10)^2}{3}}$$

Similar, the quadratic mean of the estimation error is defined for the second row (a_{21}, a_{22}, a_{23}). The plot shows us that the estimation error is small and decreases as the sample size increases. We also check this behavior under different types of

generating distributions, including

$$X_T^{raw} \sim \begin{pmatrix} N(0,1) \\ \chi_1^2 \end{pmatrix}$$
$$X_S \sim \begin{pmatrix} N(0,1) \\ \chi_1^2 \end{pmatrix}$$
$$X_T = \begin{pmatrix} 1 & 2 & 10 \\ 2 & 1 & -20 \end{pmatrix} \begin{pmatrix} X_T^{raw} \\ 1 \end{pmatrix}$$

We call this model 2 and observe a similar trend as model 1 as shown in Fig. 2.4(c).

Applications to Alzheimer's disease study

We demonstrate the application of the framework to correct distributional shift between two Alzheimer's disease (AD) datasets and show how such a strategy can lead to improved pooled data analysis. The two datasets come from the Alzheimer's Disease Neuroimage Initiative (ADNI) project and Wisconsin Alzheimer's Disease Research Center (W-ADRC). Both studies follow similar protocols for acquiring CSF samples from participants and measuring protein levels Vanderstichele et al. (2012). It is known that the CSF protein levels are indicative of neurofibrillary tangles and amyloid plaques, characteristic of AD pathology. The distributions of the protein measurements across the two datasets are different due to various reasons described in the literature Vanderstichele et al. (2012), which makes pooled analysis and/or transferring results from one dataset to the other problematic. For example, a threshold derived for the ADNI dataset may not be applicable to the W-ADRC dataset. Both datasets included eight distinct CSF protein levels measured on seven proteins ($A\beta_{1\text{-}42}$ is measured by two methods), where the distributional shift needs to be corrected. In both W-ADRC and ADNI, the measured proteins include $A\beta_{1\text{-}38}$, $A\beta_{1\text{-}40}$, $A\beta_{1\text{-}42}$, p-tau_{181}, t-*tau*, NFL and neurogranin. While W-ADRC dataset provides 125 samples, ADNI includes 284 samples (see Table 2.1). After correcting the distributional shift, we fit statistical models which include age, sex, and CSF

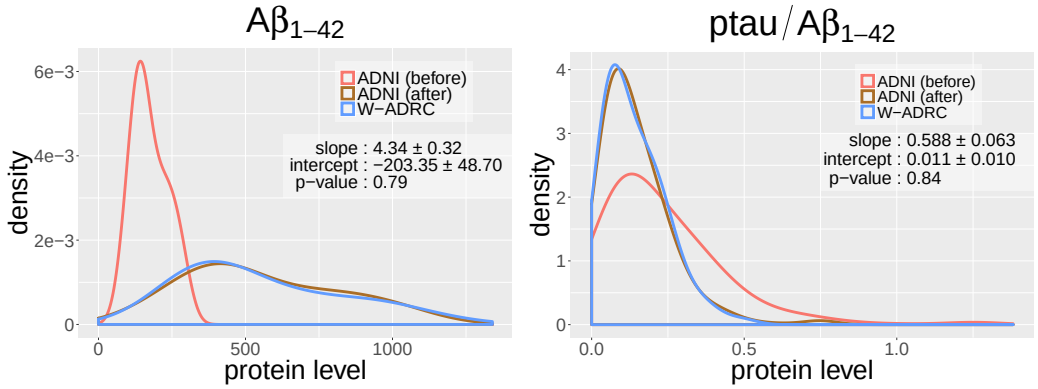
Table 2.1: Variations of age and diagnosis status across datasets.

	ADNI	W-ADRC
Sample size	284	125
Age range ($55 \sim 65/65 \sim 75/75 \sim 85$)	$11/43/46(\%)$	$44/34/22(\%)$
Diagnosis status (CN/AD)	$60/40(\%)$	$76/24(\%)$

proteins as covariates. As a response variable, we use hippocampus volume or diagnosis status. Here, besides correcting the CSF protein levels across the two datasets, we also correct distribution shift of hippocampus volumes since they may be calculated with different image acquisition characteristics and potentially different software (Freesurfer in ADNI versus FIRST/FSL in W-ADRC). Our workflow involves three tasks **1)** correct distributional shift across the datasets for CSF protein levels, **2)** transform thresholds in ADNI to W-ADRC, **3)** pool the data together to predict the response variable (hippocampus volume, diagnosis status) within regression or classification.

Correct distributional shift of CSF

Table 2.1 shows that the age distributions as well as the proportion of participants who are healthy (CN) and diseased (AD) in the two datasets are not exactly the same, which makes directly attempting a distributional shift correction in the CSF measures not very meaningful. But when other variations (confounders) co-exist together with distributional shift, as discussed earlier, we should check whether there exists a set of variables Z satisfying conditions given in Thm. 2.2. We previously described how choosing $Z = \{D, \text{age}\}$ satisfies Thm.2.2. Such a Z is also the minimal d-separating set. To proceed with the analysis, we divide our samples in $d = 6$ groups based on all possible combinations of diagnosis status (AD/CN) and age ranges ($55 \sim 65/65 \sim 75/75 \sim 85$). We can now run the subsampling-MMD algorithm ($\mathcal{SSP}$) with $Z = \{D, \text{age}\}$ (iterations $B = 2000$) to correct the distributional shift in X. We show two representative results in Fig. 2.5. For each plot in Fig. 2.5, depending on the subsamples randomly collected from 10 iterations, we plot the distributions of protein levels and a protein ratio measure (widely used in the



(a) (b)

Figure 2.5: The plots of (a) $A\beta_{1\text{-}42}$ and (b) p-*tau*/$A\beta_{1\text{-}42}$ show the empirical distributions of W-ADRC samples (blue), ADNI samples (red) and transformed ADNI samples (brown). W-ADRC samples are nicely matched with transformed ADNI samples.

aging/AD literature) in ADNI before/after correction (red/brown) with respect to W-ADRC baseline (blue). We see that the distributions of raw measures are very different between ADNI (using the AlzBio3 xMAP assay) and W-ADRC (using the ELISA INNOTEST assay). After our correction, the distributions are matched for all 8 CSF protein measurements and both protein ratios that are relevant in AD research (p-*tau*/$A\beta_{1\text{-}42}$ and t-*tau*/$A\beta_{1\text{-}42}$). We randomly select one iteration and apply the hypothesis test, which accepts the transformations with high p-values. We also use the infinitesimal Jackknife to estimate the standard deviations of parameters and report them in Fig. 2.5.

Transferring thresholds for disease staging across datasets

After performing our correction, CSF protein measurements across the two datasets can be analyzed together. We can evaluate the effect of using models (or thresholds) derived for the ADNI dataset on W-ADRC by transferring the criteria directly. For example, five CSF based biomarker signature (thresholds) developed for AD using ADNI participants Shaw et al. (2009) can now be transferred to the W-ADRC

Table 2.2: The performance of thresholds in ADNI and W-ADRC.

W-ADRC	t-*tau*	$A\beta_{1\text{-}42}$	p-*tau*	$\frac{\text{t-}tau}{A\beta_{1\text{-}42}}$	$\frac{\text{p-}tau}{A\beta_{1\text{-}42}}$
Threshold	568.08	629.39	48.86	0.77	0.07
Sensitivity(%)	75.86	89.66	82.75	93.10	93.10
Specificity(%)	92.23	69.90	67.96	86.41	79.61

ADNI	t-*tau*	$A\beta_{1\text{-}42}$	p-*tau*	$\frac{\text{t-}tau}{A\beta_{1\text{-}42}}$	$\frac{\text{p-}tau}{A\beta_{1\text{-}42}}$
Threshold	93.00	192.00	23.00	0.39	0.10
Sensitivity(%)	69.6	96.4	67.9	85.7	91.1
Specificity(%)	92.3	76.9	73.1	84.6	71.2

The W-ADRC thresholds are derived from corresponding ADNI thresholds reported in the literature Shaw et al. (2009) using Algorithm 2.

dataset. Given a threshold for any specific CSF protein, we can evaluate a sample in W-ADRC by comparing the corresponding measurements with the transformed threshold. The procedure produces sensitivity and specificity (for detection of AD) for each of the 8 CSF protein measurements and the 2 derived ratios. Our final thresholds, sensitivities and specificities based on the experiments are shown in Tab. 2.2. The accuracy estimates suggest that all derived thresholds work well — we find that the sensitivity and specificity are competitive with the results reported for ADNI Shaw et al. (2009) (cf. Table 4), and show how results/models from one dataset may be transferable to another dataset using our proposal.

Pooling and analyzing the two datasets together

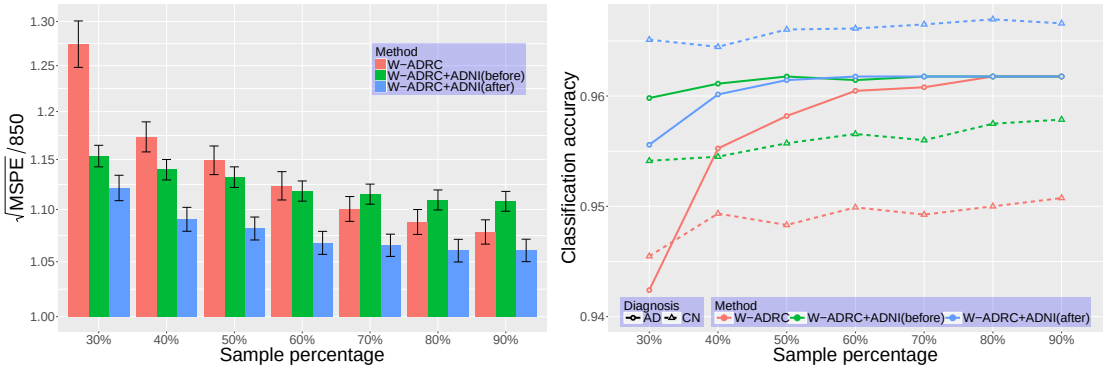
For the final experiment, we evaluate whether predictors from both datasets can be pooled for predicting hippocampus volume and diagnosis status (response variables) within regression and classification. We build a linear regression model based on age, sex and CSF proteins (*after* distributional shift correction) to identify associations with hippocampus volume. In order to evaluate the accuracy of the model, we randomly choose 25 samples (20%) from W-ADRC data to serve as the test set. For evaluation purposes, we generate three different types of training datasets: using W-ADRC samples only, W-ADRC plus raw (uncorrected) ADNI

samples and W-ADRC plus transformed ADNI samples. Note that the data used to generate the training set is based on all 284 ADNI samples and the remaining 100 W-ADRC samples. To obtain prediction errors for each of the three schemes with respect to varying training sample sizes, we vary the training sample size by choosing b% samples from each of the two datasets and then change b from 30% to 90% in 10% increments. To avoid performance variation due to random choice of samples, after the test set is chosen, we run 5 bootstraps to select training set and fit the model. Finally, we run 80 bootstraps to generate multiple test sets and evaluate the model performance. In this way, based on 400 bootstraps, we are able to obtain a more stable prediction error and are able to calculate the standard deviation. The square root of mean squared prediction error (MSPE) scaled by a constant is shown in Fig. 2.6(a). We can see that the prediction errors decrease as training sample size increases, while the W-ADRC plus transformed ADNI data consistently offers the best performance.

Next, the same setup is used to predict AD status with a Support Vector Machine (SVM) classifier. Because the ratio of AD and CN is biased in the test set from W-ADRC, we set a uniform prior in SVM and separately report the classification accuracy for participants with AD and without AD in Fig. 2.6(b).

2.6 Discussion

There is growing interest in the design of infrastructure and platforms that allow scientists across different sites and even continents to contribute scientific data and explore scientific hypotheses that cannot be evaluated on smaller datasets. Such efforts can be facilitated via the availability of theory and algorithms to identify whether pooling is meaningful, how the data should be harmonized and later, how statistically meaningful and reproducible scientific conclusions can be obtained. We described a statistical framework that addresses some of the natural issues that arise in this regime, in particular, providing conditions where distributional shift between data sets can be corrected. The experimental results suggest promising potential applications of this idea in aging and Alzheimer's disease (AD) studies.



(a) (b)

Figure 2.6: Plot (a) shows the trend of mean squared prediction error (MSPE) for hippocampus volume as the sample size increases using 400 bootstraps. The bar plot covers the prediction error for three types of training set as depicted in the legend, including using W-ADRC only (red), W-ADRC plus ADNI (green) and W-ADRC plus transformed ADNI (blue). The third model continues to perform the best. Plot (b) shows the trend of classification accuracy with respect to patients with AD (solid line) and healthy patients (dotted line) as sample size increases using 400 bootstraps. An SVM model is used and three types of training set are shown in the legend. For samples with AD, the three methods converge to the same accuracy as the training sample size increases. For healthy controls, the W-ADRC plus the transformed ADNI dataset is always better than the other two schemes. It is interesting to see that W-ADRC plus the raw ADNI data also performs better than W-ADRC alone, possibly because only 25(24%) subjects from W-ADRC are diagnosed with AD – with few AD samples, even the uncorrected ADNI data nicely informs the classification model.

We presented the framework for pooling and analyzing datasets from multiple heterogeneous sources. For an almost ideal framework, the biases across multiple datasets may decrease but continue to exist. That leads to the question that how we can decide whether the pooling is beneficial or not. In the next Chapter, we are going to discuss hypothesis tests and theory on when it is beneficial to pool multi-source datasets.

3 HYPOTHESIS TESTS AND THEORETICAL ANALYSIS ON WHEN POOLING MULTI-SOURCE DATASETS IS BENEFICIAL

3.1 Introduction

In this Chapter, we discuss the hypothesis tests and theory on when pooling multi-source datasets in beneficial. In the last two decades, statistical machine learning algorithms for processing massive datasets have been intensively studied for a wide-range of applications in computer vision, biology, chemistry and healthcare Murdoch and Detsky (2013); Tarca et al. (2007). While the challenges posed by large scale datasets are compelling, we pointed out in Chapter 2 that one is often faced with a fairly distinct set of technical issues for studies in biological and health sciences. The statistical analyses in these studies are often underpowered for the sample sizes available, and necessitates efforts to identify similar datasets elsewhere so that the combined sample size of the "pooled" dataset is enough to determine significant associations between a response and a set of predictors, e.g., within linear regression. In Chapter 2, we presented a framework to combine multiple heterogeneous datasets together. However, there might exist other biases between two datasets beyond what we considered, and there might exist a few remaining biases after we transform the datasets by our framework. In this Chapter, we consider the problem whether it is beneficial to combine multiple datasets when biases exist. This raises several fundamental technical questions. When is it meaningful to pool datasets for estimating a simple statistical model (e.g., linear regression)? When can we guarantee improvements in statistical power, and when are such pooling efforts not worth it? Can we give a hypothesis test and obtain p-values to inform our policies/decisions? While related problems have been studied in machine learning from an algorithm design perspective, even simple hypothesis tests which can be deployed by a researcher in practice, are currently unavailable. Our goal is to remove this significant limitation.

The realization that "similar" datasets from multiple sites can be pooled to

potentially improve statistical power is not new. With varying empirical success, models tailored to perform regression in multi-site studies Group (2002), Haase et al. (2009), Klunk et al. (2015) have been proposed, where due to operational reasons, recruitment and data acquisition are distributed over multiple sites, or even countries. When the pooling is being performed *retrospectively* (i.e., after the data has been collected), resolving site-specific confounds, such as distributional shifts or biases in measurements, is essential before estimation/inference of a statistical model. We will *not* develop new algorithms for estimating the distributional mismatch or for performing multi-site regression — rather, our primary goal is to identify the regimes (and give easily computable checks) where this regression task on a pooled dataset is statistically meaningful, assuming that good pre-processing schemes are available. We will present a rigorous yet simple to implement hypothesis test, analyze its behavior, and show extensive experimental evidence (for an important scientific problem). The practitioner is free to use his/her preferred procedure for the "before step" (estimating the distributional shifts).

Our contributions.

i) Our main result is a hypothesis test to evaluate whether pooling data across multiple sites for regression (before or after correcting for site-specific distributional shifts) can improve the estimation (mean squared error) of the relevant coefficients (while permitting an influence from a set of confounding variables).

ii) We derive analogous results in the high-dimensional setting by leveraging a different set of analysis techniques. Using an existing sparse multi-task Lasso model, we show how the utility of pooling can be evaluated even when the support set of the features (predictors) is not exactly the same across sites using ideas broadly related to high dimensional simultaneous inference Dezeure et al. (2015). We show ℓ_2-consistency rate, which supports the use of sparse multi-task Lasso when sparsity patterns are not totally identical.

iii) On an important scientific problem of analyzing early Alzheimer's disease (AD) individuals, we provide compelling experimental results showing consistent acceptance rate and statistical power. Via a package in CRAN/R, this will directly facilitate many multi-site regression analysis efforts in the short to medium term future.

Related Work

Meta-analysis approaches. If datasets at multiple different sites *cannot* be shared or pooled, the task of deriving meaningful scientific conclusions from *results of multiple independently conducted analyses* generally falls under the umbrella term of "meta analysis". The literature provides various strategies to cumulate the general findings from analyses on different datasets. But even experts believe that, minor violations of assumptions can lead to misleading scientific conclusions Greco et al. (2013), and substantial personal judgment (and expertise) is needed to conduct them. It is widely accepted that when the ability to pool the data is an option, simpler schemes may perform better.

Domain adaptation/shift. Separately, the idea of addressing "shift" within datasets has been rigorously studied within statistical machine learning, see Patel et al. (2015); Li (2012). For example, domain adaptation, including dataset and covariate shift, seeks to align (the distributions of) multiple datasets to enable follow-up processing Ben-David and Schuller (2003). Typically, such algorithms assume a bias in the sampling process, and adopt re-weighting as the solution Huang et al. (2007b); Gong et al. (2013). Alternatively, a family of such methods assume that sites (or datasets) differ due to feature distortions (e.g., calibration error), which are resolved, in general, by minimizing some distance measure between appropriate distributions Baktashmotlagh et al. (2013); Pan et al. (2011); Long et al. (2015). In general, these approaches have nice theoretical properties Ben-David et al. (2010); Cortes and Mohri (2011); Zhou et al. (2016). However, it is important to note that the domain adaptation literature *focuses on the algorithm itself* – to resolve the distributional site-wise differences. It does *not* address the issue of whether

pooling the datasets, after applying the calculated adaptation (i.e., transformation), is beneficial. Our goal in this Chapter is to assess whether multiple datasets can be pooled — either *before* or usually *after* applying the best domain adaptation methods — for improving our estimation of the relevant coefficients within linear regression. We propose a hypothesis test to directly address this question.

The high-dimensional case. Neuroimaging scenarios, in general, involve predicting a response (e.g., cognitive score) from high dimensional predictors such as imaging scans and genetic data, which in general, entails Lasso-type formulations unlike the classical regression models. Putting multi-task representation learning Maurer et al. (2016),Ando and Zhang (2005),Maurer et al. (2013) together with a sparsity regularizer, we get multi-task Lasso model Liu et al. (2009); Kim and Xing (2010). Although this seems like a suitable model Chen et al. (2012), it assumes that multiple tasks (sites here) have an identical active set of predictors. Instead, we find that sparse multi-task Lasso Lee et al. (2010), roughly, a multi-task version of sparse group Lasso Simon et al. (2013); Lee et al. (2010) is a better starting point. There is no theoretical analysis in Simon et al. (2013); although a ℓ_2-consistency for sparse group lasso is derived in Chatterjee et al. (2012) using a general proof procedure for M-estimators, it does not take into account the specific sparse group Lasso properties, thereby, making the result non-informative for sparse group Lasso (much less, sparse multi-task Lasso). Specifically, as we will see shortly, in sparse multi-task Lasso, the joint effects of two penalties induces a special type of asymmetric structure. We show a new result, in the style of Lasso Meinshausen and Yu (2009); Liu and Zhang (2009), for ℓ_2 convergence rate for this model. It matches with results known for Lasso and group Lasso, and identifies regimes where the sparse multi-task (multi-site) setting is advantageous.

Simultaneous High dimensional Inference. Simultaneous high dimensional inference models such as multi sample-splitting and de-biased Lasso is an active research topic in statistics Dezeure et al. (2015). Multi sample-splitting use half of the dataset for variable selection and the rest for calculating p-values. De-biased Lasso chooses one feature as a response and the others as predictors to estimate a Lasso model; this procedure is repeated for each feature. Estimators from De-biased

Lasso asymptotically follow the multi-normal distribution Dezeure et al. (2016), and using Bonferroni-Holm adjustment produces simultaneous p-values. Such ideas together with the ℓ_2-convergence results for sparse multitask Lasso, will help extend our analysis to the high dimensional setting.

3.2 Hypothesis Test for Multi-Site Regression

We first describe a simple setting where one seeks to apply standard linear regression to data pooled from multiple sites. For presentation purposes, we will deal with variable selection issues later. Within this setup, we will introduce our main result — a hypothesis test to evaluate statistical power improvements (e.g., mean squared error) when running a regression model on a pooled dataset. We will see that the proposed test is transparent to the use of adaptation algorithms, if any, to pre-process the multi-site data. In later sections, we will present convergence analysis and extensions to the large p setting. Matrices (vectors/scalars) are upper case (and lower case). $\|.\|_*$ is the nuclear norm.

We first introduce the single-site regression model. Let $X \in \mathbb{R}^{n\times p}$ and $y \in \mathbb{R}^{n\times 1}$ denote the feature matrix of predictors and the response vector respectively. If β corresponds to the coefficient vector (i.e., predictor weights), then the regression model is

$$\min_{\beta} \quad \frac{1}{n}\|y - X\beta\|_2^2 \tag{3.1}$$

where $y = X\beta^* + \epsilon$ and $\epsilon \sim N(0, \sigma^2)$ I.I.D. if β^* is the true coefficient vector from which y is generated. The mean-squared error (MSE) and ℓ_2-consistency of regression is well-known. The mean-squared error (MSE) of (3.1) is $\mathbb{E}\|\hat{\beta} - \beta^*\|_2^2 = \tilde{r}\left((X^TX)^{-1}\right)\sigma^2$. If k denotes the number of sites, then one may first apply a domain adaptation scheme to account for the distributional shifts between the k different predictors $\{X_i\}_{i=1}^k$, and then run a regression model. If the underlying "concept" (i.e., predictors and responses relationship) can be assumed to be the same across the different sites, then it is reasonable to impose the *same* β for all sites. For instance,

as discussed in Section 3.1, the influence of protein measurements on cognitive scores of an individual is assumed to be invariant to demographics. Nonetheless, if the distributional mismatch correction is imperfect, we may define $\Delta\beta_i = \beta_i - \beta^*$ where $i \in \{1, \ldots, k\}$ as the residual difference between the site-specific coefficients and the true shared coefficient vector (in the ideal case, we have $\Delta\beta_i = 0$). In the multi-site setting, we can write

$$\min_{\beta} \sum_{i=1}^{k} \tau_i^2 \|y_i - X_i\beta\|_2^2 \tag{3.2}$$

where for each site i we have $y_i = X_i\beta^* + X_i\Delta\beta_i + \epsilon_i$ and $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$ i.i.d. Here, τ_i is a weighting parameter for each site, if such information is available.

Our main goal is to test if the combined regression improves the estimation for a single site. We can pose this question in terms of improvements in the mean squared error (MSE). Hence, w.l.o.g. using site 1 as the reference, we have the following reduced objective by setting $\beta_1 = \beta^*$ and $\tau_1 = 1$ in (3.2),

$$\min_{\beta} \|y_1 - X_1\beta\|_2^2 + \sum_{i=2}^{k} \tau_i^2 \|y_i - X_i\beta\|_2^2 \tag{3.3}$$

Clearly, when the sample size is not large enough, the multi-site formulation in (3.3) may reduce variance significantly, because of the averaging effect in the objective function, while increasing the bias by a little. This reduces the Mean Squared Error (MSE), see Figure 3.1. Note that while traditionally, the unbiasedness property was desirable, an extensive body of literature on ridge regression suggests that the quantity of interest should really be $\mathbb{E}\|\hat{\beta} - \beta^*\|_2^2$. These ideas are nicely studied within papers devoted to the "bias-variance" trade-off. Similar to these results, we will focus on the mean squared error because the asymptotic consistency properties that come with an unbiased estimator are not meaningful here anyway — the key reason we want to pool

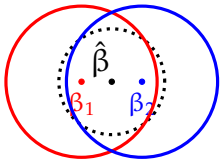


Figure 3.1: β_1 and β_2 are 1^{st} and 2^{nd} site coefficients. After combination, β_1's bias increases but variance reduces, resulting in a smaller MSE.

datasets in the first place is because of small sample sizes. We now provide a result showing how the tuning parameters $\tau_2, \ldots, \tau_k$ can be chosen.

Theorem 3.1. $\tau_i = \frac{\sigma_1}{\sigma_i}$ *achieve the smallest variance in* $\hat{\beta}$.

Remarks: This result follows from observing that the each site's contribution is inversely proportional to site-specific noise level, σ_i. We will show that this choice of τ_is also leads to a simple mechanism to setup a hypothesis test.

Sharing all βs

In the specification above, the estimates of β_i across all k sites are restricted to be the same. Without this constraint, (3.3) is equivalent to fitting a regression *separately* on each site. So, a natural question is whether this constraint improves estimation. To evaluate whether MSE is reduced, we first need to quantify the change in the bias and variance of (3.3) compared to (3.1). To do so, we introduce a few notations. Let n_i be the sample size of site i, and let $\hat{\beta}_i$ denote the regression estimate from a specific site i. We have $\Delta\hat{\beta}_i = \hat{\beta}_i - \hat{\beta}_1$. We define the length kp vector $\Delta\beta^T$ as $\Delta\beta^T = (\Delta\beta_2^T, ..., \Delta\beta_k^T)$ (similarly for $\Delta\hat{\beta}^T$). We use $\hat{\Sigma}_i$ to give the sample covariance matrix of data (predictors) from the site i and $G \in \mathbb{R}^{(k-1)p\times(k-1)p}$ is the covariance matrix of $\Delta\hat{\beta}$, where $G_{ii} = (n_1\hat{\Sigma}_1)^{-1} + (n_i\tau_i^2\hat{\Sigma}_i)^{-1}$ and $G_{ij} = (n_1\hat{\Sigma}_1)^{-1}$ whenever $i \neq j$.

Let the difference in bias and variance between the single site model in (3.1) and the multi-site model in (3.3) be $\mathtt{Bias}_\beta$ and $\mathtt{Var}_\beta$ respectively. Let $\hat{\Sigma}_2^k = \sum_{i=2}^k n_i\tau_i^2\hat{\Sigma}_i$ and $\hat{\Sigma}_1^k = n_1\hat{\Sigma}_1 + \hat{\Sigma}_2^k$. We have,

Lemma 3.2. *For model* (3.3)*, we have*

$$\frac{\|\mathtt{Bias}_\beta\|_2^2}{\|G^{-1/2}\Delta\beta\|_2^2} \leqslant \|(\hat{\Sigma}_1^k)^{-2}(\hat{\Sigma}_2^k(n_1\hat{\Sigma}_1)^{-1}\hat{\Sigma}_2^k + \hat{\Sigma}_2^k)\|_*, \tag{3.4}$$

$$\mathtt{Var}_\beta = \sigma_1^2 \left\|(n_1\hat{\Sigma}_1)^{-1} - (n_1\hat{\Sigma}_1 + \hat{\Sigma}_2^k)^{-1}\right\|_*. \tag{3.5}$$

Remarks: The above result bounds the increase in bias and the reduction in variance (see discussion of Figure 3.1). Since our goal is to test MSE reduction — in principle, we can use bootstrapping to calculate MSE approximately. This procedure has a significant computational footprint. Instead, (3.4) (which comes from a one-step Cauchy-Schwartz), gives a *sufficient condition for MSE reduction* as shown below.

Theorem 3.3. ***a)*** *Model* (3.3) *has smaller MSE of* $\hat{\beta}$ *than model* (3.1) *whenever*

$$H_0 : \|G^{-1/2}\Delta\beta\|_2^2 \leqslant \sigma_1^2. \tag{3.6}$$

b) *Further, we have the following test statistic,*

$$\left\| \frac{G^{-1/2}\Delta\hat{\beta}}{\sigma_1} \right\|_2^2 \sim \chi^2_{(k-1)*p} \left(\left\| \frac{G^{-1/2}\Delta\beta}{\sigma_1} \right\|_2^2 \right), \tag{3.7}$$

where $\|G^{-1/2}\Delta\beta/\sigma_1\|_2$ *is called a "condition value".*

Remarks: This is our main test result. Although σ_i is typically unknown, it can be easily replaced using its site-specific estimation. Theorem 3.3 implies that we can conduct a non-central χ^2 distribution test based on the statistic. Also, (3.6) shows that the non-central χ^2 distribution, which the test statistics will follow, has a non-central parameter smaller than 1 when the sufficient condition H_0 holds. Meanwhile, in obtaining the (surprisingly simple) sufficient condition H_0, no other arbitrary assumption is needed except the application of Cauchy-Schwartz. From a practical perspective, Theorem 3.3 implies that the sites, in fact, do not even need to share the whole dataset to assess whether pooling will be useful. Instead, the test only requires *very high-level* information such as $\hat{\beta}_i$, $\hat{\Sigma}_i$, σ_i and n_i for all participating sites – which can be transferred very cheaply with no additional cost of data storage, or privacy implications. The following result deals with the special case where we have two participating sites.

Corollary 3.4. *For the case where we have two participating sites, the condition* (3.6) *from*

Theorem 3.3 reduces to

$$H_0 : \Delta\beta^T((n_1\hat{\Sigma}_1)^{-1} + (n_2\tau_2^2\hat{\Sigma}_2)^{-1})^{-1}\Delta\beta \leqslant \sigma_1^2. \tag{3.8}$$

Remarks: The left side above relates to the Mahalanobis distance between β_1, β_2 with covariance $(n_1\hat{\Sigma}_1)^{-1} + (n_2\tau_2^2\hat{\Sigma}_2)^{-1}$, implying that the test statistic is a type of a normalized metric between the two regression models.

Sharing a subset of βs

In numerous pooling scenarios, we are faced with certain systemic differences in the way predictors and responses associate across sites. For example, socio-economic status may (or may not) have a significant association with a health outcome (response) depending on the country of the study (e.g., due to insurance coverage policies). Unlike in Section 3.2, we now relax the restriction that all coefficients are same across sites, see Figure 3.2. The model in (3.3) will now include another design matrix of predictors $Z \in R^{n\times q}$ and a corresponding coefficients γ_i for each site i,

$$\min_{\beta,\gamma} \quad \sum_{i=1}^{k} \tau_i^2 \|y_i - X_i\beta - Z_i\gamma_i\|_2^2 \tag{3.9}$$

$$y_i = X_i\beta^* + X_i\Delta\beta_i + Z_i\gamma_i^* + \epsilon_i,\ \tau_1 = 1 \tag{3.10}$$

Our goal is still to evaluate whether the MSE of β reduces. We do not take into

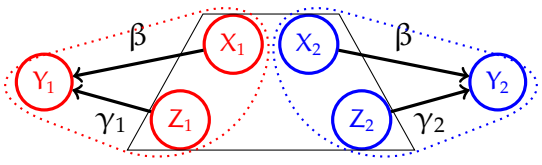


Figure 3.2: X and Z influence the response Y. After adjustment, X_1 and X_2 may be close requiring same β. However, Z_1 and Z_2 may differ a lot, and we need different γ_1 and γ_2.

account the MSE change in γ because they correspond to site-specific variables. For estimating them, $\hat{\beta}$ can first be computed from (3.9). Treating it as fixed entity now, $\hat{\gamma}_i$ can be computed using y_i and Z_i on each site independently. Clearly, if $\hat{\beta}$ is close to the "true" β^*, it will also enable a better estimation of site-specific variables. It turns out that, if $\hat{\Sigma}_i$s are replaced by the conditional covariance, the analysis from Section 3.2 still holds for this case. Specifically, let $\hat{\Sigma}_{ab_i}$ be the sample covariance matrix between features a and b from some site i. We have,

Theorem 3.5. *Analysis in Section 3.2 holds for* β *in* (3.9) *by replacing* $\hat{\Sigma}_i$ *with* $\tilde{\Sigma}_i = \hat{\Sigma}_{xx_i} - \hat{\Sigma}_{xz_i}(\hat{\Sigma}_{zz_i})^{-1}\hat{\Sigma}_{zx_i}$

Remarks: The test now allows evaluating statistical power improvements focused on the subset of the coefficient vector that is shared and permits site-specific confounds, enabling much flexibility in practice. For example, we can test which subset of parameters might benefit from parameter estimation on pooled data from multiple sites.

3.3 Pooling in High Dimensional Regression

We now describe our analysis of pooling multi-site data in the high-dimensional setting where $p \gg n$. The key challenge here is that *variable section* has to be a first order concern. In classical regression, the ℓ_2 consistency properties are well known and so our focus in Section 3.2 was devoted entirely to deriving sufficient conditions for the hypothesis test. In other words, imposing the same β across sites works in (3.3) because we understand its consistency. In contrast, here, one cannot enforce a shared β for all sites *before* the active set of predictors within each site are selected — directly imposing the same β leads to a serious loss of ℓ_2-consistency, making follow-up analysis problematic. Therefore, once a suitable model for high-dimensional multi-site regression is chosen, the first requirement is to characterize its consistency.

We start with the multi-task Lasso (a special case of group Lasso) Liu et al. (2009), where the authors show that the strategy selects better explanatory features

compared to separately fitting Lasso on each site. But this algorithm underperforms when the sparsity pattern of the predictors is not identical across sites, so we use a recent variant called sparse multi-task Lasso Lee et al. (2010) – essentially substituting "sites" for "tasks". The sparse *multi-site* Lasso in $p \gg n$ setting (p is the number of predictors) is given as

$$\hat{B}^{\lambda} = \arg\min_{\beta} \sum_{i=1}^{k} \|y_i - X_i \beta_i\|_2^2 + \lambda \Lambda(B) \tag{3.11}$$

$$\Lambda(B) = \alpha \sum_{j=1}^{p} \|\beta^j\|_1 + (1-\alpha)\sqrt{k} \sum_{j=1}^{p} \|\beta^j\|_2, \tag{3.12}$$

where λ is the Lasso regularization parameter. Here, $B \in R^{k \times p}$ is a matrix where the i^{th} row corresponds to the coefficients from i^{th} site (k sites in total). Also, β_i with subscript denotes the i^{th} row (site) of B, we use β^j with superscript to give the j-th column (coefficients) of B. The hyper-parameter $\alpha \in [0, 1]$ balances the two penalties; a larger α weighs the ℓ_1 penalty more and a smaller α puts more weight on the grouping. This will play an important role for the remainder of this section. Similar to a Lasso-based regularization parameter, λ here will produce a solution path (to select coefficients) for a given α. We first address the consistency behavior of the sparse multi-site Lasso in (3.11), which was not known in the literature.

ℓ_2 consistency

Our analysis of (3.11) is related to known results for Lasso Meinshausen and Yu (2009) and the group Lasso Liu and Zhang (2009). Recall that $X_1, \ldots, X_k$ are the data matrices from k sites. We define $\bar{n} = \max_{i=1}^{k}\{n_i\}$ and $C = \bar{n}^{-1}\textsc{diag}(X_1^T X_1, ..., X_k^T X_k)$ where $\textsc{diag}(A, B)$ corresponds to constructing a block-diagonal matrix with A and B as blocks on the diagonal. We require the following useful properties of C ($\|\cdot\|_0$ denotes ℓ_0-norm).

Definition 3.6. *The m-sparse minimal and maximal eigenvalues of C, denoted by $\phi_{\min}(m)$*

and $\phi_{\max}(m)$, *are*

$$\min_{v:\|v\|_0 \leqslant \lceil m \rceil} \frac{v^T C v}{v^T v} \quad \textit{and} \quad \max_{v:\|v\|_0 \leqslant \lceil m \rceil} \frac{v^T C v}{v^T v} \tag{3.13}$$

Let us call a feature "active" if its coefficient is non-zero. We know that each site may have different active features: let $s_h \leqslant kp$ be the sum of the number of active features over all sites. Similarly, s_p is the cardinality of the union of features that are active in at least one site ($s_h \leqslant k s_p, s_p \leqslant p$). Recall that when $\alpha \neq 0$, we add the Lasso penalty to the multi-site Lasso penalty. Whenever the sparsity patterns are assumed to be similar across all sites, α is small. On the other hand, to encourage site-specific sparsity patterns, we may set α to be large. The following two technical results analyze these cases independently.

Theorem 3.7. *Let* $0 \leqslant \alpha \leqslant 0.4$. *Assume there exist constants* $0 \leqslant \rho_{\min} \leqslant \rho_{\max} \leqslant \infty$ *such that*

$$\begin{aligned} &\liminf_{n\to\infty} \phi_{\min}\left(s_p \log \bar{n}\left(1 + \frac{2\alpha}{1-2\alpha}\right)^2\right) \geqslant \rho_{\min} \\ &\limsup_{n\to\infty} \phi_{\max}(s_p + \min\{\sum_{i=1}^{k} n_i, kp\}) \leqslant \rho_{\max}. \end{aligned} \tag{3.14}$$

Then, for $\lambda \propto \sigma\sqrt{\bar{n}\log(kp)}$, *there exists a constant* $\omega > 0$ *such that, with probability converging to 1 for* $n \to \infty$,

$$\frac{1}{k}\|\hat{B}^\lambda - B^*\|_F^2 \leqslant \omega\sigma^2 \frac{\bar{s}\log(kp)}{\bar{n}}, \tag{3.15}$$

where $\bar{s} = \{(1-\alpha)\sqrt{s_p} + \alpha\sqrt{s_h/k}\}^2$, σ *is the noise level.*

Remarks: The above result agrees with known results for multi-task Lasso Liu et al. (2009); Liu and Zhang (2009) when the sparsity patterns are the same across sites. The simplest way to interpret Theorem 3.7 is via the ratio $r = \frac{s_h}{s_p}$. Here, $r = k$ when the sparsity patterns are the same across sites. As r decreases, the sparsity patterns across sites start to differ, in turn, the sparse multi-site Lasso from (3.11) will provide stronger consistency compared to the multi-site Lasso (which

corresponds to $\alpha = 0$). In other words, whenever we expect site-specific active features, the ℓ_2 consistency of (3.11) will improve as one includes an additional ℓ_1-penalty together with multi-site Lasso.

Observe that for the non-sparse β^j, we can verify that $\|\beta^j\|_1$ and $\sqrt{k}\|\beta^j\|_2$ have the same scale. On the other hand, for sparse β^j, $\|\beta^j\|_1$ has the same scale as $\|\beta^j\|_2$, i.e., with no $\sqrt{k}$ penalization. Unlike Theorem 3.7 where the sparsity patterns across sites are similar, due to this scaling issue, the parameters α and λ need to be 'corrected' for the setting where sparsity patterns have little overlap. We denote this corrected versions by $\tilde{\alpha} = \frac{\alpha}{(1-\alpha)\sqrt{k}+\alpha}$ and $\tilde{\lambda} = ((1-\alpha)\sqrt{k} + \alpha)\lambda$.

Theorem 3.8. *Let* $0.4 \leqslant \tilde{\alpha} \leqslant 1$. *Assume there exist constants* $0 \leqslant \rho_{\min} \leqslant \rho_{\max} \leqslant \infty$ *such that*

$$\begin{aligned} &\liminf_{n\to\infty} \phi_{\min}\left(s_h \log \bar{n}\left(1 + \frac{(1-\tilde{\alpha})}{\tilde{\alpha}}\right)^2\right) \geqslant \rho_{\min} \\ &\limsup_{n\to\infty} \phi_{\max}(s_h + \min\{\sum_{i=1}^{k} n_i, kp\}) \leqslant \rho_{\max}. \end{aligned} \tag{3.16}$$

Then, for $\tilde{\lambda} \propto \sigma\sqrt{\bar{n}\log(kp)}$, *there exists* $\omega > 0$ *such that, with probability converging to 1 for* $n \to \infty$, *we have* (3.15) *with* $\tilde{s} = \{(1-\tilde{\alpha})\sqrt{s_p/k} + \tilde{\alpha}\sqrt{s_h/k}\}^2$ *instead of* $\bar{s}$.

Remarks: This result agrees with known results for Lasso Meinshausen and Yu (2009) when the sparsity patterns are completely different across sites. In this case (i.e., α is large), the sparse multi-site Lasso has stronger consistency compared to Lasso ($\alpha = 1$). The sparse multi-site Lasso is preferable as $r = \frac{s_h}{s_p}$ increases. Note that although $\tilde{\alpha}$ and $\tilde{\lambda}$ are used for the technical results instead of α and λ, in practice, one can simply scale the chosen αs appropriately. For instance, with $k = 100$, we see that $\alpha \approx 0.99$ corresponds to a $\tilde{\alpha} = 0.95$. We do not penalize by $\sqrt{k}$ when the sparsity patterns across sites share few of the features. To see this, first observe that when sparsity patterns are similar, most of the groups we have are non-sparse, and the effects of $\sqrt{k}\|\beta^j\|_2$ and $\|\beta^j\|_1$ have the same scale. This is simply because, $\sqrt{k}\sqrt{a_1^2 + ... + a_k^2}$ is close to $|a_1| + ... + |a_k|$ whenever $|a_1|, ..., |a_k|$ are close. However when sparsity patterns across sites share few features only, most of

the groups are going to be sparse. For these groups, we should use $\|\beta^j\|_2$, because in this setting $\sqrt{a_1^2 + 0 + ... + 0}$ is close to $|a_1| + 0 + ... + 0$.

Performing hypothesis tests. Theorems 3.7 and 3.8 show consistency of sparse multi-site Lasso estimation. Hence, if the hyper-parameters α and λ are known, we can estimate the coefficients B^*. This variable selection phase can be followed by a hypothesis test, similar to Theorem 3.3 from Section 3.2. The only remaining issue is the choice of α and existing methods suggest a heuristic. They set it to 0.05 when it is known that sparsity patterns are similar across sites and 0.95 otherwise Simon et al. (2013). Joint cross-validation for α and λ is shown to perform worse Simon et al. (2013). Below, we instead provide a data-driven alternative that works well in practice.

Choosing α using simultaneous inference. Our theoretical results in Thm. 3.7 (and Thm. 3.8 resp.) seem to suggest that increasing (and decreasing resp.) α will always improve consistency; however, this ends up requiring much stronger m-sparsity conditions. We now describe a procedure to choose α. First, recall that an active feature corresponds to a variable with non-zero coefficient. We call a feature (or predictor) "site-active" if it is active at a site, an "always-active" feature is active at all k sites. The proposed solution involves three steps. **(1)** First, we apply simultaneous inference (like multi sample-splitting or de-biased Lasso) using all features at each of the k sites with FWER control. This step yields "site-active" features for each site, and therefore, gives the set of always-active features and the sparsity patterns. **(2)** Then, each site runs a Lasso and chooses a λ_i based on cross-validation. We then set $\lambda_{\text{multi-site}}$ to be the minimum among the best λ's from each site. Using $\lambda_{\text{multi-site}}$, we can vary α to fit various sparse multi-site Lasso models – each run will select some number of always-active features. We plot α versus the number of always-active features. **(3)** Finally, based on the sparsity patterns from the *site-active* set, we can estimate whether the sparsity patterns across sites are similar or different (i.e., share few active features). Then, based on the plot from step (2), if the sparsity patterns from the site-active sets are different (similar) across

sites, then the smallest (largest) value of α that selects the minimum (maximum) number of always-active features is chosen.

3.4 Experiments

Our experiments are two-fold. First we perform simulations evaluating the hypothesis test from Section 3.2 and the sparse multi-site Lasso from Section 3.3. We then conduct an experiment to pool two Alzheimer's disease (AD) datasets coming from different ongoing studies to evaluate improvements in power, and checking whether the proposed tests provide insights into the regimes when pooling is beneficial for regression, thereby yielding tangible statistical benefits, in neuroscience/neuroimaging research.

Power and Type I Error of Theorem 3.3. The first set of simulations evaluate the setting from Section 3.2 where the coefficients are same across two different sites. The inputs for the two sites are set as $X_1, X_2 (\in \mathbb{R}^{n \times 3}) \sim \mathcal{N}(0, \Sigma)$ with $\Sigma = 0.5(I + E)$ (where I is identity and E is a 3×3 matrix of 1s). The true coefficients are given by $\beta_1 \sim U(0, 4I)$ and $\beta_2 = \beta_1 + 0.1$ (where $U(\cdot)$ is multivariate uniform), and the noise corresponds to $\epsilon_1 \sim \mathcal{N}(0, 3I)$ and $\epsilon_2 \sim \mathcal{N}(0, 0.5I)$ for the two sites respectively. With this, the responses are set as $y_1 = X_1\beta_1 + \epsilon_1$ and $y_2 = X_2\beta_2 + \epsilon_2$. Using $\{X_1, y_1\}$ and $\{X_2, y_2\}$, the *shared* $\hat{\beta}$ are estimated. The simulation is repeated 100 times with 9 different sample sizes ($n = 2^b$ with $b = 4, \ldots, 12$) for each repetition. Figure 3.3(a) shows the MSE of two-site (blue bars) and a baseline single-site (red bars) model computed using the corresponding $\hat{\beta}$s on first site. Although both MSEs decrease as n increases, the two-sites model consistently produces smaller MSE – with large gains for small sample sizes (left-end of Figure 3.3(a)). Figure 3.3(d) shows the acceptance rates of our proposed hypothesis test (from (3.6) and (3.8)) with 0.05 significance level. The purple solid line is the sufficient condition from Theorem 3.3, while the dotted line is where the MSE of the baseline single-site model starts to decrease below that of two-site model. The trend in Figure 3.3(d) implies that as n increases, the test tends to reject pooling the multi-site data with power $\to 1$.

Further, the type I error is well-controlled to the left of the solid line, and is low between the two lines.

Power and Type I Error of Theorem 3.5. The second set of simulations evaluate the confounding variables setup from Section 3.2. Similar to Section 3.4, here we have $(X_1, Z_1), (X_2, Z_2) \sim \mathcal{N}(0, \Sigma)$ with $\Sigma = \begin{pmatrix} 0.5I_{3\times3} + 0.5E_{3\times3}, & 0.2E_{3\times5} \\ 0.2E_{5\times3}, & 0.8I_{5\times5} + 0.2E_{5\times5} \end{pmatrix}$. β_1 and β_2 are the same as earlier. $\gamma_1 = (1, 1, 2, 2, 2)^T$ and $\gamma_2 = (2, 2, 2, 1, 1)^T$ are the coefficients for Z_1 and Z_2 respectively. The new responses y_1 and y_2 will have the extra terms $Z_1\gamma_1$ and $Z_2\gamma_2$ respectively. Figure 3.3(b,e) shows the results. All the observations from Figure 3.3(a,d) hold here as well. For small n, MSE of two-site model is much smaller than baseline, and as sample size increases this difference reduces. The test accepts with high probability for small n, and as sample size increases it rejects with high power. The regimes of low type I error and high power in Figure 3.3(e) are similar to those from Figure 3.3(d).

Sparse multi-sites Lasso ℓ_2-consistency

We now use 4 sites with $n = 150$ samples each and $p = 400$ features to test the sparse multi-site model from Section 3.3. We set the design matrices X_i ($i = 1, \ldots, 4$) $\sim \mathcal{N}(0, \Sigma)$ with $\Sigma = 0.8I_{p\times p} + 0.2E_{p\times p}$. The two cases where sparsity patterns are shared, and not shared, are considered separately.

Few sparsity patterns shared. 6 shared features and 14 site-specific features (out of the 400) are set to be active in 4 sites. Each of shared features is sampled from $U(0, 4)$ for first two sites and $U(0, 0.5)$ for the rest. All the site-specific features are $\sim U(0, 4)$. The noise $\epsilon_i \sim \mathcal{N}(0, 1)$, and the responses are $y_i = X_i\beta_i + \epsilon_i$. Figure 3.3(c) shows the 10-fold cross validation error as λ changes (i.e., solution path) for different α settings, including the value from our proposed selection procedure (from Section 3.3), Lasso ($\alpha = 1$), group Lasso ($\alpha = 0$) and arbitrary values $\alpha = 0.05$, 0.95 (as suggested by Simon et al. (2013)). Our chosen $\alpha = 0.97$ (the blue curve in Figure 3.3(c)) has smallest error, across all the λs, thereby implying a better ℓ_2

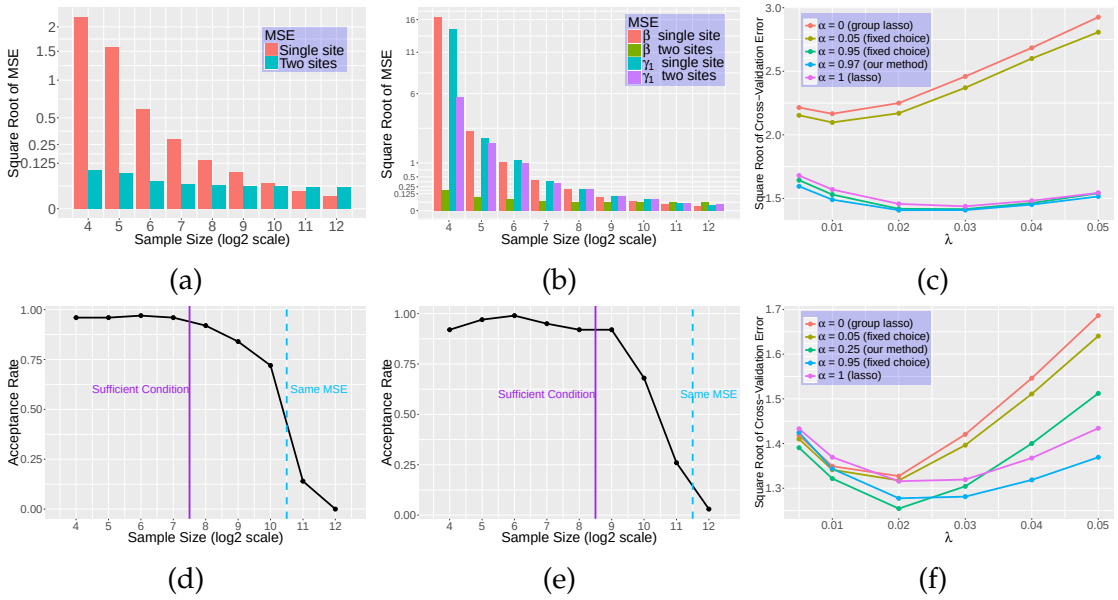


Figure 3.3: (a,d) $\hat{\beta}$'s MSE and the acceptance rate, (b,e) MSE of $\hat{\beta}$ and $\hat{\gamma}_1$, and the acceptance rate. These use 100 bootstrap repetitions. Solid line in (d,e) is when the condition from Theorem 3.3 is 1. Dotted line is when MSE of single-site and multi-site models are the same. (c) λ solution path when sparsity patterns are dissimilar across sites, (f) The alternate regime where sparsity patters are similar

consistency. We show that $\alpha = 0.97$ discovers more always-active features, while preserving the ratio of correctly discovered active features to all the discovered ones.

Most sparsity patterns shared. Unlike the earlier case, here we set 16 shared and 4 site-specific features (both $\sim \mathrm{U}(0, 4)$) to be active among all the 400 features. The result, shown in Figure 3.3(f), is similar to Figure 3.3(c). The proposed choice of $\alpha = 0.25$ competes favorably with alternate choices while preserving the correctly discovered number of always-active features. Unlike the previous case, the ratio of correctly discovered active features to all discovered ones increases here.

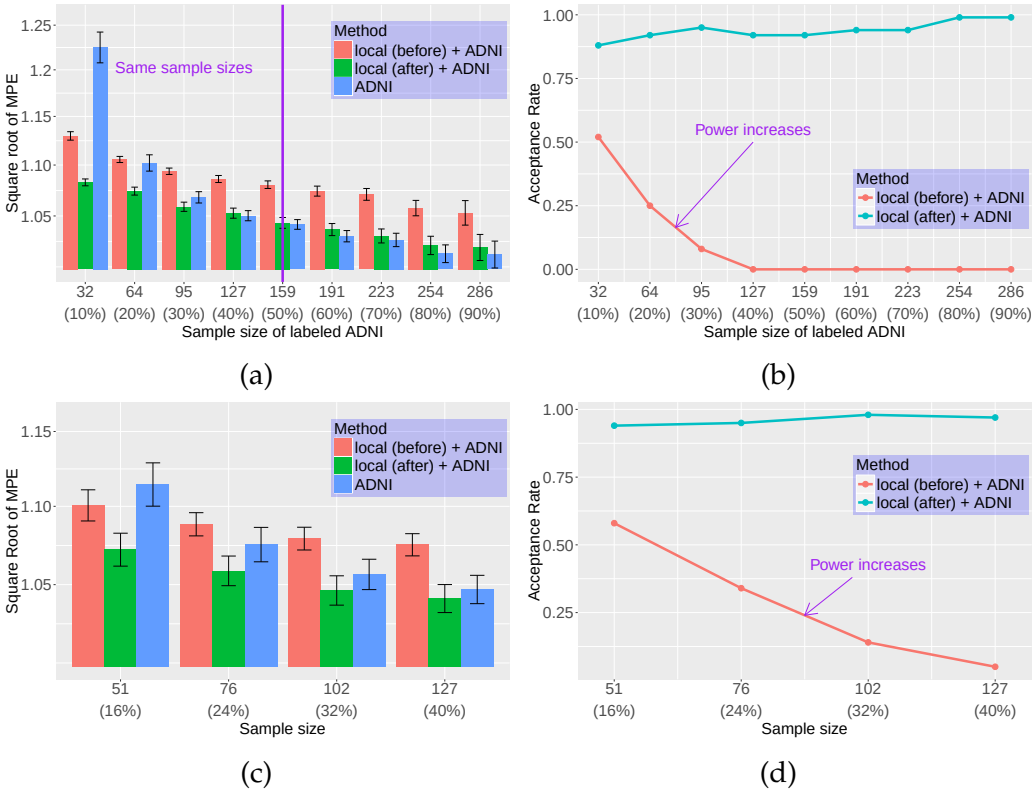


Figure 3.4: (a,c) MPE for the pooled regression model (green) compared to baselines plotted against training subset size of ADNI. x-axis is number/fraction of ADNI labeled samples used in training (apart from ADlocal). (b,d) show the acceptance rates for (a,c). Unlike in (a), (c) restricts same training data size for ADNI and ADlocal.

Combining AD datasets from multiple sites

We now evaluate whether two AD datasets acquired at different sites – a Alzheimer's Disease Neuroimage Initiative (ADNI) dataset and a local dataset (ADlocal) – can be combined. The two datasets we use are – an open-source Alzheimer's Disease Neuroimage Initiative (ADNI) dataset, and a local dataset (ADlocal). ADNI is an open consortium with the goal of understanding AD related cognitive decline, and in the process, develop clinical interventions aimed at delaying the disease onset. ADlocal corresponds to a recent (smaller) initiative local study for the AD related

decline. We used 318 samples from ADNI and 156 samples from ADlocal. The input variables are 8 Cerebrospinal fluid (CSF) protein levels, and the response is hippocampus volume. The CSF proteins are “1-38-Tr", “1-40-Tr", “1-42-Tr", “NFL", "AB42", “$\mathtt{htau}$", “$\mathtt{ptau}_{181}$", and “Neurogranin". The two datasets have different age and diagnosis distributions, and hence, we subsample 81 samples from either of sites to control age and diagnosis variation. Using these 81 samples from each dataset, we perform domain adaptation (using a maximum mean discrepancy objective as a measure of distance between the two marginals) and transform CSF proteins from ADlocal to match ADNI. The transformed data is then used to evaluate whether adding ADlocal data to ADNI will improve the regression performed on the ADNI data. This is done by training a regression model on the ‘transformed’ ADlocal and a subset of ADNI data, and then testing the resulting model on the remaining ADNI samples. We use two baseline models each of which are trained using – ADNI data *alone*; and *non-transformed* ADlocal (with ADNI subset).

Figure 3.4(a,b) show the resulting mean prediction error (MPE) scaled by the estimated noise level in ADNI responses, and the corresponding acceptance rate (with significance level 0.05) respectively. The x-axis in Figure 3.4(a,b) represents the size of ADNI subset used for training. As expected, the MPE reduces as this subset size increases. Most importantly, pooling after transformation (green bars) seems to be the most beneficial in terms of MPE reduction. As shown in Figure 3.4(a), to the left of purple line where the subset size is smaller than ADlocal datasize, pooling the datasets improves estimation. This is the small sample size regime which necessitates pooling efforts in general. As the dataset size increases (to the right of x-axis in Figure 3.4(a)) the resulting MPE for the pooled model is close to what we will achieve using the ADNI data by itself.

Since pooling after transformation is at least as good as using ADNI data alone, our proposed hypothesis test accepts the combination with high rate ($\approx 95\%$) as can be seen from Figure 3.4(b). The test rejects the pooling strategy with high power for combining before domain adaptation (see Figure 3.4(b)), as one would expect. This rejection power increases rapidly as sample size increases as pointed out on the red curve in Figure 3.4(b). The results in Figure 3.4(c,d) show the setting where one

cannot change the dataset sizes at the sites i.e., the training set uses an equal number of labeled samples from both the ADNI and ADlocal (x-axis in Figure 3.4(c)), and the testing set always corresponds to 20% of ADNI data. This is a more interesting scenario for a practitioner compared to Figure 3.4(a,b), because in Figure 3.4(c,d) we use the same sample sizes for both datasets. The trends in Figure 3.4(c,d) are the same as Figure 3.4(a,b).

Extra experiments on synthetic data

We present the hypothesis test simulation when $p = 6$ in Fig. 3.5, which is similar to the simulations done in Fig. 3.3. However, here the dimension p of β is 6 instead of 3.

Table 3.1: Add multi-sites Lasso on Lasso.

α	0	0.05	0.95	0.97 (our)	1
CDR	0.1423	0.1463	0.2747	0.2863	0.2955
CDV	78	78	75	75	73
CDG	5	5	3	3	1

Table 3.2: Add Lasso on multi-sites Lasso.

α	0	0.05	0.25 (our)	0.95	1
CDR	0.2292	0.2381	0.2453	0.2841	0.2885
CDV	80	80	79	75	73
CDG	16	16	15	11	11

For sparse multi-Sites lasso simulation, we report correctly discovered number of active variables (CDV), ratio of CDV and total number of discovered variables (CDR), and correctly discovered number of always-active features (CDG).
From Table3.1 and Table 3.2 we see that our chosen α helps sparse multi-sites

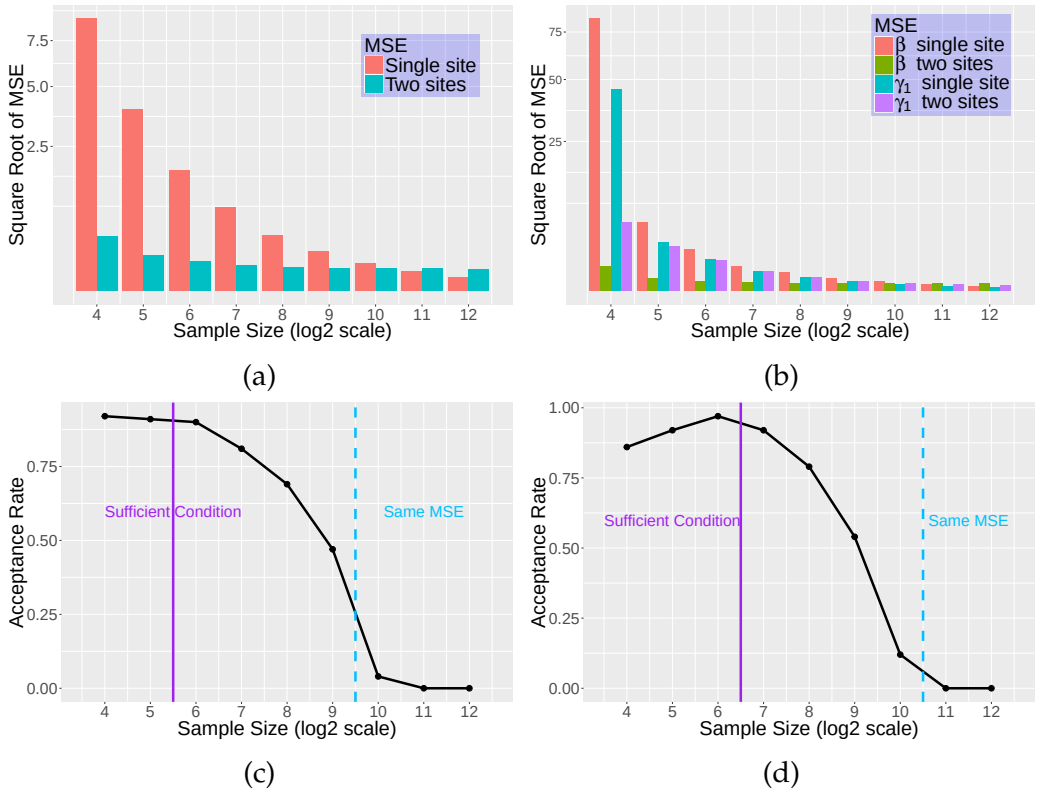
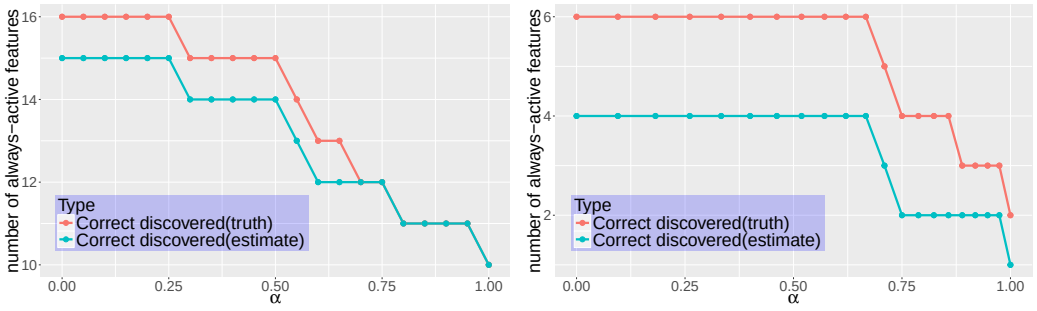


Figure 3.5: The figure is similar to the simulations done in Figure 3.3. However, here the dimension p of β is 6 instead of 3. (a,c) are MSE of $\hat{\beta}$ and the corresponding acceptance rate of our hypothesis test (from Section2.1). (b,d) are MSE of $\hat{\beta}$ and $\hat{\gamma}_1$ and the corresponding acceptance rate (from Section2.2). These are based on 100 bootstrap repetitions. The solid line in (c,d) represents the point where the condition from Theorem 3.3 is equal to 1. The dotted line is when MSE of $\hat{\beta}$ is the same for single-site and multi-site models.

Lasso to discover more or preserve always-active features. The number and rate of correctly discovered number of active variables given by our chosen α are also among the best.

We show an example for choosing α in Fig. 3.6. We here point out a caveat about our choice of α when sparsity patterns share few features and always-active



(a) (b)

Figure 3.6: These plots show that *site-active* set from simultaneous inference provides information of always-active features (which is then used to choose the hyper-parameters α an λ). In (a), we add Lasso on multi-sties Lasso, and $\alpha = 0.25$ is chosen. Similarly, in (b), we add multi-sites Lasso on Lasso, and $\alpha = 0.97$ is chosen.

features exist. In this setting, we do want to discover more always-active features. Hence, we decrease α from 1 and stop at the point where we just select one more always-active feature.

3.5 Discussion

In this work, we present a hypothesis test to answer the question of whether pooling multiple datasets acquired from different sites is guaranteed to increase statistical power for regression models. For both low and high dimensional linear regression, we precisely identify regimes where such pooling is sensible, and show how such policy decisions can be made via simple checks executable on each site before any data transfer ever happens. We also show empirical results by combining two Alzheimer's disease datasets in the context of different regimes proposed by our analysis, and showing that the regression fit improves as suggested by the theory.

We show the hypothesis tests and theory on when pooling multi-source datasets is beneficial. It the next Chapter, we study a different problem and consider that data is spatio-temporal, limited and lie in high dimension.

4 NON-PARAMETRIC SPARSE ADDITIVE AUTO-REGRESSIVE NETWORK MODELS AND THEORETICAL ANALYSIS FOR SPATIO-TEMPORAL DATA

4.1 Introduction

In this Chapter, we consider data is spatio-temporal ,limited and lie in high dimension. We derive a non-parametric sparse additive auto-regressive network model and study it's theoretical behavior. Multi-variate time series data arise in a number of settings including neuroscience (Brown et al. (2004); Ding et al. (2011)), finance (Rydberg and Shephard (1999)), social networks (Chavez-Demoulin and McGill (2012); Aït-Sahalia et al. (2010); Zhou et al. (2013)) and others (Heinen (2003); Matteson et al. (2011); Ogata (1999)). A fundamental question associated with multi-variate time series data is to quantify influence between different players or nodes in the network (e.g. how do firing events in one region of the brain trigger another, how does a change in stock price for one company influence others, e.t.c). To address such a question requires *estimation of an influence network* between the d players or nodes. Two challenges that arise in estimating such an influence network are (i) developing a suitable network model; and (ii) providing theoretical guarantees for estimating such a network model when the number of nodes d is large.

Prior approaches for addressing these challenges involve parametric approaches (Fokianos and Tjøstheim (2011); Fokianos et al. (2009); Hall et al. (2016)). In particular, Hall et al. (2016) use a generalized linear model framework for estimating the high-dimensional influence network. More concretely, consider samples $(X_t)_{t=0}^T$ where $X_t \in \mathbb{R}^d$ for every t which could represent continuous data, count data, binary data or others. We define $p(.)$ to be an exponential family probability distribution, which includes, for example, the Gaussian, Poisson, Bernoulli and others to handle different data types. Specifically, $x \sim p(\theta)$ means that the distribution of the scalar x is associated with the density $p(x|\theta) = h(x)\exp[\varphi(x)\theta - Z(\theta)]$, where

$Z(\theta)$ is the so-called log partition function, $\varphi(x)$ is the sufficient statistic of the data, and $h(x)$ is the base measure of the distribution. For the prior parametric approach in Hall et al. (2016), the j^{th} time series observation of X_{t+1} has the following model:

$$X_{t+1,j}|X_t \sim p\left(\nu_j + \sum_{k=1}^{d} A^*_{j,k} X_{t,k}\right),$$

where $A^* \in \mathbb{R}^{d\times d}$ is the network parameter of interest. Theoretical guarantees for estimating A^* are provided in Hall et al. (2016). One of the limitations of parametric models is that they do not capture non-linear effects such as saturation. Non-parametric approaches are more flexible and apply to broader network model classes but suffer severely from the curse of dimensionality (see e.g. Stone (1985)).

To overcome the curse of dimensionality, the sparse additive models (SpAM) framework was developed (see e.g. Koltchinskii and Yuan (2010); Meier et al. (2009); Raskutti et al. (2012); Ravikumar et al. (2010)). Prior approaches based on the SpAM framework have been applied in the regression setting. In this Chapter, we consider samples generated from a *non-parametric sparse additive auto-regressive model*, generated by the generalized linear model (GLM),

$$X_{t+1,j}|X_t \sim p\left(\nu_j + \sum_{k=1}^{d} f^*_{j,k}(X_{t,k})\right), \tag{4.1}$$

where $f^*_{j,k}$ is an unknown function belonging to a reproducing kernel Hilbert space $\mathcal{H}_{j,k}$. The goal is to estimate the d^2 functions $(f^*_{j,k})_{1\leqslant j,k\leqslant d}$.

Prior theoretical guarantees for sparse additive models have focused on the setting where samples are independent. In this Chapter, we analyze the convex penalized sparse and smooth estimator developed and analyzed in Koltchinskii and Yuan (2010); Raskutti et al. (2012) under the dependent Markov chain model (4.1). To provide theoretical guarantees, we assume the Markov chain "mixes" using concepts of β and ϕ-mixing of Markov chains. In particular, in contrast to the para-

metric setting, our mean-squared error is a function of β or ϕ mixing co-efficients, and the smoothness of the RKHS function class. We also support our theoretical guarantees with simulations and show through simulations and a performance analysis on real data the potential advantages of using our non-parametric approach.

Our contributions. As far as we are aware, we are the first to provide a theoretical analysis of high-dimensional non-parametric auto-regressive network models. In particular, we make the following contributions.

i) We provide a scalable non-parametric framework using technologies in sparse additive models for high-dimensional time series models that capture non-linear, non-parametric framework. This provides extensions to prior work on high-dimensional parametric models by exploiting RKHSs.

ii) In Section 4.4, we provide the most substantial contribution which is an upper bound on mean-squared error that applies in the high-dimensional setting. Our rates depend on the sparsity of the function, smoothness of each univariate function, and mixing co-efficients. In particular, our mean-squared error upper bound scales as:

$$\max\left(\frac{s\log d}{\sqrt{mT}}, \sqrt{\frac{m}{T}}\tilde{\epsilon}_m^2\right),$$

up to logarithm factors, where s is the maximum degree of a given node, d is the number of nodes of the network, T is the number of time points. Here $\tilde{\epsilon}_m$ refers to the univariate rate for estimating a single function in RKHS with m samples (see e.g. Raskutti et al. (2012)) and $1 \leqslant m \leqslant T$ refers to the number of *blocks* needed depending on the β and ϕ-mixing co-efficients. If the dependence is weak and $m = O(T)$, our mean-squared error bounds are optimal up to log factors as compared to prior work on independent models Raskutti et al. (2012) while if dependence is strong $m = O(1)$, we obtain the slower rate (up to log factors) of $\frac{1}{\sqrt{T}}$ that is optimal under no dependence assumptions.

iii) We also develop a general proof technique for addressing high-dimensional time series models. Prior proof techniques in Hall et al. (2016) rely heavily on parametric assumptions and constraints on the parameters which allow us to use martingale concentration bounds. This proof technique explicitly exploits mixing co-efficients which relies on the well-known "blocking" technique for sequences of dependent random variables (see e.g. Mohri and Rostamizadeh (2010); Nobel and Dembo (1993)), which does not require parametric assumptions. In the process of the proof, we also develop upper bounds on Rademacher complexities for RKHSs and other empirical processes under mixing assumptions rather than traditional independence assumptions as discussed in Section 4.5.

iv) In Section 4.6, we demonstrate through both a simulation study and real data example the flexibility and potential benefit of using the non-parametric approach. In particular we show improved prediction error performance on higher-order polynomials applied to a Chicago crime dataset.

The remainder of this Chapter is organized as follows: In Section 4.2, we introduce the preliminaries for RKHSs, and beta-mixing of Markov chains. In Section 4.3, we present the non-parametric multi-variate auto-regressive network model and the sparse and smooth estimation scheme. In Section 4.4, we present the main theoretical results and focus on specific cases of finite-rank kernels and Sobolev spaces. In Section 4.5, we provide the main steps of the proof, deferring the more technical steps to the appendix and in Section 4.6, we provide a simulation study that supports our theoretical guarantees and a performance analysis on Chicago crime data.

4.2 Preliminaries

In this section, we introduce the basic concepts of RKHSs and standard definitions of β and ϕ mixing for stationary processes.

Reproducing Kernel Hilbert Spaces

First we introduce the basics of RKHSs and smoothness assumptions. Given a subset $\mathcal{X} \subset \mathbb{R}$ and a probability measure $\mathbb{Q}$ on $\mathcal{X}$, we consider a Hilbert space $\mathcal{H} \subset \mathcal{L}^2(\mathbb{Q})$, meaning a family of functions $g : \mathcal{X} \to \mathbb{R}$, with $\|g\|_{\mathcal{L}^2(\mathbb{Q})} < \infty$, and an associated inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ under which $\mathcal{H}$ is complete. The space $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS) if there exists a symmetric function $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ such that: (a) for each $x \in \mathcal{X}$, the function $\mathcal{K}(x, \cdot)$ belongs to the Hilbert space $\mathcal{H}$, and (b) we have the reproducing relation $g(x) = \langle g(\cdot), \mathcal{K}(x, \cdot) \rangle_{\mathcal{H}}$ for all $g \in \mathcal{H}$. This function $\mathcal{K}$ is the so-called kernel function, which under suitable regularity conditions, has an eigen-expansion of the form

$$\mathcal{K}(x, x') = \sum_{i=1}^{\infty} \mu_i \Phi_i(x) \Phi_i(x')$$

guaranteed by Mercer's theorem Mercer (1909), where $\mu_1 \geqslant \mu_2 \geqslant \mu_3 \geqslant ... \geqslant 0$ are a non-negative sequence of eigenvalues, and $\{\Phi_i\}_{i=1}^{\infty}$ are the associated eigenfunctions, taken to be orthonormal in $\mathcal{L}^2(\mathbb{Q})$. As has already been established (see e.g. Aronszajn (1950); Koltchinskii and Yuan (2010); Meier et al. (2009); Raskutti et al. (2012); Smola and Schölkopf (1998); Wahba (1990)), these eigenvalues play a crucial role in our analysis, since they ultimately determine the univariate rate $\epsilon_m, \tilde{\epsilon}_m$ (to be specified later) for estimating a single function in RKHS.

Since the eigenfunctions $\{\Phi_i\}_{i=1}^{\infty}$ form an orthonormal basis, any function $g \in \mathcal{H}$ has an expansion of the form $g(x) = \sum_{i=1}^{\infty} a_i \Phi_i(x)$, where $a_i = \langle g, \Phi_i \rangle_{\mathcal{L}^2(\mathbb{Q})} = \int_{\mathcal{X}} g(x) \Phi_i(x) d\mathbb{Q}(x)$ are (generalized) Fourier coefficients. For any two functions in $\mathcal{H}$, say $g(x) = \sum_{i=1}^{\infty} a_i \Phi_i(x)$ and $f(x) = \sum_{i=1}^{\infty} b_i \Phi_i(x)$, we can define two distinct inner products. The first is the usual inner product in the space $\mathcal{L}^2(\mathbb{Q})$-namely, $\langle g, f \rangle_{\mathcal{L}^2(\mathbb{Q})} := \int_{\mathcal{X}} g(x) f(x) d\mathbb{Q}(x)$. By Parseval's theorem, it has an equivalent repre-

sentation in terms of the expansion coefficients, namely

$$\langle g, f \rangle_{\mathcal{L}^2(\mathbb{Q})} = \sum_{i=1}^{\infty} a_i b_i.$$

The second inner product, denoted $\langle g, f \rangle_{\mathcal{H}}$, is the one that defines the Hilbert space which can be written in terms of the kernel eigenvalues and generalized Fourier coefficients as

$$\langle g, f \rangle_{\mathcal{H}} = \sum_{i=1}^{\infty} \frac{a_i b_i}{\mu_i}.$$

For more background on reproducing kernel Hilbert spaces, we refer the reader to various standard references Aronszajn (1950); Saitoh (1988); Smola and Schölkopf (1998); Wahba (1990); Weinert (1982).

Furthermore, for a subset $S_j \in \{1, 2, .., d\}$, let $f_j := \sum_{k \in S_j} f_{j,k}(x_k)$, where $x_k \in \mathcal{X}$ and $f_{j,k} \in \mathcal{H}_{j,k}$ is the RKHS that $f_{j,k}$ lies in. Hence we define the norm

$$\|f_j\|^2_{\mathcal{H}_j(S_j)} := \sum_{k \in S_j} \|f_{j,k}\|^2_{\mathcal{H}_{j,k}},$$

where $\|\cdot\|_{\mathcal{H}_{j,k}}$ denotes the norm on the univariate Hilbert space $\mathcal{H}_{j,k}$.

Mixing

Now we introduce standard definitions for dependent observations based on mixing theory Doukhan (1994) for stationary processes.

Definition 4.1. *A sequence of random variables* $Z = \{Z_t\}_{t=0}^{\infty}$ *is said to be* stationary *if for any* t_0 *and non-negative integers* t_1 *and* t_2*, the random vectors* $(Z_{t_0}, ..., Z_{t_0+t_1})$ *and* $(Z_{t_0+t_2}, ..., Z_{t_0+t_1+t_2})$ *have the same distribution.*

Thus the index t or time, does not affect the distribution of a variable Z_t in a stationary sequence. This does not imply independence however and we capture the dependence through mixing conditions. The following is a standard definition

giving a measure of the dependence of the random variables Z_t within a stationary sequence. There are several equivalent definitions of these quantities, we are adopting here a version convenient for our analysis, as in Mohri and Rostamizadeh (2010); Yu (1994).

Definition 4.2. *Let $Z = \{Z_t\}_{t=0}^{\infty}$ be a stationary sequence of random variables. For any $i_1, i_2 \in Z \cup \{0, \infty\}$, let $\sigma_{i_1}^{i_2}$ denote the σ-algebra generated by the random variables $Z_t, i_1 \leqslant t \leqslant i_2$. Then, for any positive integer ℓ, the β-mixing and ϕ-mixing coefficients of the stochastic process Z are defined as*

$$\beta(\ell) = \sup_t E_{B \in \sigma_0^t}[\sup_{A \in \sigma_{t+\ell}^{\infty}} |P[A|B] - P[A]|],$$

$$\phi(\ell) = \sup_{t, A \in \sigma_{t+\ell}^{\infty}, B \in \sigma_0^t} |P[A|B] - P[A]|.$$

Z is said to be β-mixing (ϕ-mixing) if $\beta(\ell) \to 0$ (resp. $\phi(\ell) \to 0$) as $\ell \to \infty$. Furthermore Z is said to be algebraically *β-mixing (algebraically ϕ-mixing) if there exist real numbers $\beta_0 > 0$ (resp. $\phi_0 > 0$) and $r > 0$ such that $\beta(\ell) \leqslant \beta_0/\ell^r$ (resp. $\phi(\ell) \leqslant \phi_0/\ell^r$) for all ℓ.*

Both $\beta(\ell)$ and $\phi(\ell)$ measure the dependence of an event on those that occurred more than ℓ units of time in the past. β-mixing is a weaker assumption than ϕ-mixing and thus includes more general non-i.i.d. processes.

4.3 Model and estimator

In this section, we introduce the sparse additive auto-regressive network model and the sparse and smooth regularized schemes that we implement and analyze.

Sparse additive auto-regressive network model

From Equation (4.1) in Section 4.1, we can state the conditional distribution explicitly as:

$$\mathbb{P}(X_{t+1}|X_t) = \prod_{j=1}^{d} h(X_{t+1,j}) \exp\{\varphi(X_{t+1,j})\,\theta_j - Z(\theta_j)\},$$
$$\theta_j = \nu_j + \sum_{k=1}^{d} f^*_{j,k}(X_{t,k}),$$

where $f^*_{j,k}$ is an unknown function belonging to a RKHS $\mathcal{H}_{j,k}$, $\nu \in [\nu_{min}, \nu_{max}]^d$ are known constant offset parameters. Recall that $Z(\cdot)$ refers to the log-partition function and $\varphi(\cdot)$ refers to the sufficient statistic. This model has the Markov and conditional independence properties, that is, conditioning on the previous data at time point $t-1$, the elements of X_t are independent of one another and X_t are independent with data before time $t-1$. We note that while we assume that ν is a known constant vector, if we assume there is some unknown constant offset that we would like to estimate, we can fold it into the estimation of f^* via appending a constant 1 column in X_t.

We assume that the data we observe is $(X_t)_{t=0}^T$ and our goal is to estimate f^*, which is constructed element-wise by $f^*_{j,k}$. However, in our setting where d may be large, the sample size T may not be sufficient even under the additivity assumption and we need further structural assumptions. Hence we assume that the network function f^* is sparse which does not have too many non-zero functions. To be precise, we define the sparse supports $(S_1, S_2, ..., S_d)$ as:

$$S_j \subset \{1, 2, ..., d\}, \text{ for any } j = 1, 2, ..., d.$$

We consider network function f^* is only non-zero on supports $\{S_j\}_{j=1}^d$, which means

$$f^* \in \mathcal{H}(S) := \{f_{j,k} \in \mathcal{H}_{j,k} | f_{j,k} = 0 \text{ for any } k \notin S_j\}.$$

The support S_j is the set of nodes that influence node j and $s_j = |S_j|$ refers to the *in-degree* of node j. In this Chapter we assume that the function matrix f^* is s-sparse, meaning that f^* belongs to $\mathcal{H}(S)$ where $|S| = \sum_{j=1}^{d} |S_j| \leqslant s$. From a network perspective, s represents the total number of edges in the network.

Sparse and smooth estimator

The estimator that we analyze in this Chapter is the standard sparse and smooth estimator developed in Koltchinskii and Yuan (2010); Raskutti et al. (2012), for each node j. To simplify notation and without loss of generality, in later statements we assume $\mathcal{H}_{j,k}$ refers to the same RKHS $\mathcal{H}$, and define $\mathcal{H}_j = \{f_j | f_j = \sum_{k=1}^{d} f_{j,k}, \text{ for any } f_{j,k} \in \mathcal{H}\}$ which corresponds to the additive function class for each node j. Further we define the *empirical norm* $\|f_{j,k}\|_T^2 := \frac{1}{T}\sum_{t=0}^{T} f_{j,k}^2(X_{t,k})$. For any function of the form $f_j = \sum_{k=1}^{d} f_{j,k}$, the $(L^2(\mathbb{P}_T), 1)$ and $(\mathcal{H}, 1)$-norms are given by

$$\|f_j\|_{T,1} = \sum_{k=1}^{d} \|f_{j,k}\|_T, \text{ and } \|f_j\|_{\mathcal{H},1} = \sum_{k=1}^{d} \|f_{j,k}\|_{\mathcal{H}}$$

respectively. Using this notation, we estimate f_j^* via a regularized maximum likelihood estimator (RMLE) by solving the following optimization problem, for any $j \in \{1, 2, .., d\}$:

$$\hat{f}_j = \arg\min_{f_j \in \mathcal{H}_j} L_1(f_j) + \lambda_T \|f_j\|_{T,1} + \lambda_H \|f_j\|_{\mathcal{H},1}, \tag{4.2}$$

where $L_1(f_j)$ is defined as

$$\frac{1}{2T}\sum_{t=0}^{T} \left(Z(\nu_j + f_j(X_t)) - (\nu_j + f_j(X_t))\varphi(X_{t+1,j})\right).$$

Here (λ_T, λ_H) is a pair of positive regularization parameters whose choice will be specified by our theory. An attractive feature of this optimization problem is that, as a straightforward consequence of the representer theorem Kimeldorf and Wahba (1971); Smola and Schölkopf (1998), it can be reduced to an equivalent convex program in $\mathbb{R}^T \times \mathbb{R}^{d^2}$. In particular, for each $(j, k) \in \{1, 2, ..., d\}^2$, let $\mathcal{K}$

denote the kernel function associated with RKHS $\mathcal{H}$ where $f_{j,k}$ belongs to. We define the collection of empirical kernel matrices $\mathbb{K}^{j,k} \in \mathbb{R}^{T\times T}$ with entries $\mathbb{K}^{j,k}_{t_1,t_2} = \mathcal{K}(X_{t_1,k}, X_{t_2,k})$. As discussed in Koltchinskii and Yuan (2010); Raskutti et al. (2012), by the representer theorem, any solution $\hat{f}_j$ to the variational problem can be expressed in terms of a linear expansion of the kernel matrices,

$$\hat{f}_j(z) = \sum_{k=1}^{d}\sum_{t=1}^{T} \hat{\alpha}_{j,k,t}\mathcal{K}(z_k, X_{t,k})$$

for a collection of weights $\{\hat{\alpha}_{j,k} \in \mathbb{R}^T \;\; (j,k) \in \{1,2,..,d\}^2\}$. The optimal weights are obtained by solving the convex problem

$$\begin{aligned}
&\hat{\alpha}_j = (\hat{\alpha}_{j,1}, ..., \hat{\alpha}_{j,d}) \\
&= \arg\min_{\alpha_{j,k}\in\mathbb{R}^T} \frac{1}{2T}\sum_{t=0}^{T}\left(Z\left(\theta_j^{\mathbb{K}}\right) - \theta_j^{\mathbb{K}}\varphi(X_{t+1,j})\right) \\
&+ \lambda_T \sum_{k=1}^{d}\sqrt{\frac{1}{T}\|\mathbb{K}^{j,k}\alpha_{j,k}\|_2^2} + \lambda_H \sum_{k=1}^{d}\sqrt{\alpha_{j,k}^T\mathbb{K}^{j,k}\alpha_{j,k}}, \\
&\text{where } \theta_j^{\mathbb{K}} = \nu_j + \sum_{k=1}^{d}\mathbb{K}^{j,k}\alpha_{j,k}.
\end{aligned}$$

This problem is a second-order cone program (SOCP), and there are various algorithms for solving it to arbitrary accuracy in polynomial time of (T, d), among them interior point methods (e.g., see the book Boyd and Vandenberghe (2004)).

Other more computationally tractable approaches for estimating sparse additive models have been developed in Meier et al. (2009); Ravikumar et al. (2010) and in our experiments section we use the package "SAM" based on the algorithm developed in Ravikumar et al. (2010). However from a theoretical perspective the sparse and smooth SOCP defined above has benefits since it is the only estimator with provably minimax optimal rates in the case of independent design (see e.g. Raskutti et al. (2012)).

4.4 Main results

In this section, we provide the main general theoretical results. In particular, we derive upper bounds on the mean-squared error

$$\|\hat{f} - f^*\|_T^2 := \frac{1}{T}\sum_{t=1}^{T} (\hat{f}(X_t) - \hat{f}(X_t))^2$$

under the assumption that the true network is s-sparse. The mean-squared error is the difference in empirical $\mathcal{L}_2(\mathbb{P}_T)$ norm between the regularized maximum likelihood estimator, $\hat{f}$, and the true generating network, f^*.

First we incorporate the smoothness of the RKHS $\mathcal{H}$. We refer to ϵ_m as the univariate rate, which depends on the eigenvalues of the RKHS. That ϵ_m is defined as the minimal value of σ, such that

$$\frac{1}{\sqrt{m}}\sqrt{\sum_{i=1}^{\infty} \min(\mu_i, \sigma^2)} \leqslant \sigma^2,$$

where $\{\mu_i\}_{i=1}^{\infty}$ are the eigenvalues in Mercer's decomposition of the kernel related to the univariate RKHS (see Mercer (1909)). In this Chapter, we define $\tilde{\epsilon}_m$ as the univariate rate in a slightly modified formula, which is the minimal value of σ, such that there exists a $M_0 \geqslant 1$ satisfying

$$\log(dT)\left\{\frac{3\log(M_0 dT)}{\sqrt{m}}\sqrt{\sum_{i=1}^{M_0} \min(\mu_i, \sigma^2)} + \sqrt{\frac{T}{m}}\sqrt{\sum_{M_0+1}^{\infty} \min(\mu_i, \sigma^2)}\right\} \leqslant \sigma^2.$$

Remark. Note that since the left side of the inequality for $\tilde{\epsilon}_m$ is always larger than it for ϵ_m, the definitions of $\tilde{\epsilon}_m$ and ϵ_m tell us that $\epsilon_m \leqslant \tilde{\epsilon}_m$. Furthermore $\tilde{\epsilon}_m$ is of

order $O(\epsilon_m \log(dT)^2)$ for finite rank kernel and kernel with decay rate $i^{-2\alpha}$. See Subsection 4.4 for more details. The modified definition $\tilde{\epsilon}_m$ allows us to extend the error bounds on $\|\hat{f} - f^*\|_T^2$ to the dependent case at the price of additional log factors. The $\log(dT)$ term is an artifact of the analysis and is required because samples are dependent, the M_0 term is needed because non-parametric functions can have infinite basis in RKHS and we require finite M_0 to apply Martingale concentration inequality.

Assumptions

We first state the assumptions in this subsection and then present our main results in the next subsection. Without loss of generality (by re-centering the functions as needed), we assume that

$$\mathbb{E}[f_{j,k}(X_{t,j})] = \int_{\mathcal{X}} f_{j,k}(x) d\mathbb{P}(x) = 0 \text{ for all } f_{j,k} \in \mathcal{H}_{j,k}, \text{ all } t.$$

Besides, for each $(j,k) \in \{1,...,d\}^2$, we make the minor technical assumptions:

- For any $f_{j,k} \in \mathcal{H}$, $\|f_{j,k}\|_{\mathcal{H}} \leqslant 1$ and $\|f_{j,k}\|_\infty \leqslant 1$.
- For any $\mathcal{H}$, the associated eigenfunctions in MercerâŁ™s decomposition $\{\Phi_i\}_{i=1}^\infty$ satisfy $\sup_x |\Phi_i(x)| \leqslant 1$ for each $i = 1, ..., \infty$.

The first condition is mild and also assumed in Raskutti et al. (2012). The second condition is satisfied by the bounded basis, for example, the Fourier basis. We proceed to the main assumptions by denoting $s_{max} = \max_j s_j$ as the maximum in-degree of the network and denoting $\mathcal{H}_\mu = \sum_{i=1}^\infty \mu_i$ as the trace of the RKHS $\mathcal{H}$.

Assumption 1 (Bounded Noise). *Let* $w_{t,j} = \frac{1}{2}(\varphi(X_{t+1,j}) - Z'(v_j + f_j^*(X_t)))$, *we assume that* $E[w_{t,j}] = 0$ *and with high probability* $w_{t,j} \in [-\log(dT), \log(dT)]$, *for any* $j \in \{1, 2, ..., d\}$, $t = 1, 2, ..., T$.

Remark. It can be checked that for (1) Gaussian link function with bounded noise or (2) Bernoulli link function, $w_{t,j} = O(1)$ with probability 1. For other generalized

linear model cases, such as (1) Gaussian link function with Gaussian noise or (2) Poisson link function under the assumption $f^*_{j,k} \leqslant 0$ for any (j,k), we have that $|w_{t,j}| \leqslant C\log(dT)$ with probability at least $1 - \exp(-c\log(dT))$ for some constants C and c (see the proof of Lemma 1 in Hall et al. (2016)).

Assumption 2 (Strong Convexity)**.** *For any* x, y *in an interval* $(\nu_{\min} - \alpha, \nu_{\max} + \alpha)$*,*

$$\vartheta\|x - y\|^2 \leqslant [Z(x) - Z(y) - Z'(y)(x - y)].$$

Remark. For the Gaussian link function, $\alpha = \infty$ and $\vartheta = 1$. For Bernoulli link function, $\alpha = (16\sqrt{\mathcal{H}_\mu} + 1)s_{\max}$ and $\vartheta = (e^{(\max(\nu_{\max}, -\nu_{\min}) + (16\sqrt{\mathcal{H}_\mu}+1)s_{\max})} + 3)^{-1}$. For Poisson link function, $\alpha = (16\sqrt{\mathcal{H}_\mu} + 1)s_{\max}$ and $\vartheta = e^{\nu_{\min} - (16\sqrt{\mathcal{H}_\mu}+1)s_{\max}}$ where recall that $s_{\max}$ is the maximum in-degree of the network.

Assumption 3 (Mixing)**.** *The sequence* $(X_t)_{t=0}^{\infty}$ *defined by the model* (4.1) *is a stationary sequence satisfying one of the following mixing conditions:*

(a) β*-mixing with* $r_\beta > 1$*.*

(b) ϕ*-mixing with* $r_\phi \geqslant 0.781$*.*

We can show a tighter bound when $r_\phi \leqslant 2$ using the concentration inequality from Kontorovich et al. (2008). The condition $r_\phi \geqslant 0.781$ arises from the technical condition in which $(r_\phi + 2) \times (2r_\phi - 1) \geqslant 2r_\phi$ (see the Proof of Lemma A.26). Numerous results in the statistical machine learning literature rely on knowledge of the β-mixing coefficient Mcdonald et al. (2011); Vidyasagar (2002). Many common time series models are known to be β-mixing, and the rates of decay are known given the true parameters of the process, for example, ARMA models, GARCH models, and certain Markov processes Mokkadem (1988); Carrasco and Chen (2002); Doukhan (1994). The ϕ-mixing condition is stronger but as we observe later allows a sharper mean-squared error bound.

Assumption 4 (Fourth Moment Assumption)**.** $E[g^4(x)] \leqslant CE[g^2(x)]$ *for some constant* C*, for all* $g \in \mathcal{F}_j := \cup_{|S_j|=s_j} H_j(S_j)$*, for any* $j \in \{1, 2, .., d\}$ *where the expectation is taken over* $\mathbb{Q}$*.*

Note that Assumption 4 is a technical assumption also required in Raskutti et al. (2012) and is satisfied under mild dependence across the covariates.

Main Theorem

Before we state the main result, we discuss the choice of tuning parameters λ_T and λ_H.

Optimal tuning parameters. Define $\gamma_m = c_1 \max\left(\epsilon_m, \sqrt{\frac{\log(dT)}{m}}\right)$, where $c_1 > 0$ is a sufficiently large constant, independent of T, s and d, and $m\gamma_m^2 = \Omega(-\log(\gamma_m))$ and $m\gamma_m^2 \to \infty$ as $m \to \infty$. $\tilde{\gamma}_m = \max(\gamma_m, \tilde{\epsilon}_m)$. The parameter m is a function of T and is defined in Thm. 4.3 and Thm. 4.4. Then we have the following optimal choices of tuning parameters:

$$\lambda_T \geqslant 8\sqrt{2}\sqrt{\frac{m}{T}}\tilde{\gamma}_m, \lambda_H \geqslant 8\sqrt{2}\sqrt{\frac{m}{T}}\tilde{\gamma}_m^2,$$

$$\lambda_T = O(\sqrt{\frac{m}{T}}\tilde{\gamma}_m), \lambda_H = O(\sqrt{\frac{m}{T}}\tilde{\gamma}_m^2).$$

Clearly it is possible to choose larger values of λ_T and λ_H at the expense of slower rates.

Theorem 4.3. *Under Assumptions 1, 2, 3 (a), and 4. Then there exists a constant C such that for each $1 \leqslant j \leqslant d$,*

$$\|\hat{f}_j - f_j^*\|_T^2 \leqslant C\frac{s_j}{\vartheta^2}\left(\frac{\log(dT)}{\sqrt{mT}} + \sqrt{\frac{m}{T}}\tilde{\epsilon}_m^2\right), \tag{4.3}$$

with probability at least

$1 - \frac{1}{T} - \left(c_2 exp(-c_3 m\gamma_m^2) + T^{-\left(\frac{1-c_0}{c_0}\right)}\right)$, *where* $m = T^{\frac{c_0 r_\beta - 1}{c_0 r_\beta}}$ *for β-mixing when $r_\beta \geqslant 1/c_0$, and c_2 and c_3 are constants. The parameter c_0 can be any number between 0 and 1.*

- Note that the term $\tilde{\epsilon}_m^2$ accounts for the smoothness of the function class, ϑ

accounts for the smoothness of the GLM loss, and m denotes the degree of dependence in terms of the number of blocks in T samples.

- In the very weakly dependent case $r_\beta \to \infty$ and $m = O(T)$, we recover the standard rates for sparse additive models $\frac{s_j \log d}{T} + s_j \tilde{\epsilon}_T^2$ (see e.g. Raskutti et al. (2012)) up to logarithm factors. In the highly dependent case $m = O(1)$, we end up with a rate proportional to $\frac{1}{\sqrt{T}}$ (up to log factors in terms of T only) which is consistent with the rates for the lasso under no independence assumptions.

- Note that we have provided rates on the difference of functions $\hat{f}_j - f_j^*$ for each $1 \leqslant j \leqslant d$. To obtain rates for the whole network function $\hat{f} - f^*$, we simply add up the errors and note that $s = \sum_{j=1}^{d} s_j$.

- To compare to upper bounds in the parametric case in Hall et al. (2016), if $m = O(T)$ and $\tilde{\epsilon}_m^2 = O(\frac{1}{m})$, we obtain the same rates. Note however that in Hall et al. (2016) we require strict assumptions on the network parameter instead of the mixing conditions we impose here.

- A larger c_0 leads to a larger m and a lower probability from the term $T^{-\frac{1-c_0}{c_0}}$.

When $r_\phi \geqslant 2$, Theorem 4.3 on β-mixing directly implies the results for ϕ-mixing. When $0.781 \leqslant r_\phi \leqslant 2$, we can present a tighter result using the concentration inequality from Kontorovich et al. (2008).

Theorem 4.4. *Under same assumptions as in Thm. 4.3, if we assume ϕ-mixing when $0.781 \leqslant r_\phi \leqslant 2$, then there exists a constant C such that for each $1 \leqslant j \leqslant d$,*

$$\|\hat{f}_j - f_j^*\|_T^2 \leqslant C \frac{s_j}{\vartheta^2} \left(\frac{\log(dT)}{\sqrt{mT}} + \sqrt{\frac{m}{T}} \tilde{\epsilon}_m^2 \right), \tag{4.4}$$

with probability at least $1 - \frac{1}{T} - c_2 exp(-c_3(m\gamma_m^2)^2)$, where $m = T^{\frac{r_\phi}{r_\phi + 2}}$ for ϕ-mixing when $0.781 \leqslant r_\phi \leqslant 2$, c_2 and c_3 are constants.

Note that $m = T^{\frac{r_\phi}{r_\phi+2}}$ is strictly larger than $m = T^{\frac{r_\phi-1}{r_\phi}}$ for $r_\phi \leqslant 2$ which is why Theorem 4.4 is a sharper result.

Examples

We now focus on two specific classes of functions, finite-rank kernels and infinite-rank kernels with polynomial decaying eigenvalues. First, we discuss finite (ξ) rank operators, meaning that the kernel function can be expanded in terms of ξ eigenfunctions. This class includes linear functions, polynomial functions, as well as any function class where functions have finite basis expansions.

Lemma 4.5. *For a univariate kernel with finite rank ξ, $\tilde{\epsilon}_m = O\left(\sqrt{\frac{\xi}{m}}\log^2(\xi d T)\right)$.*

Using Lemma 4.5 and ϵ_m calculated from Raskutti et al. (2012) gives us the following result. Note that for $T = O(m)$, we end up with the usual parametric rate.

Corollary 4.6. *Under the same conditions as Theorem 4.3, consider a univariate kernel with finite rank ξ. Then there exists a constant C such that for each $1 \leqslant j \leqslant d$,*

$$\|\hat{f}_j - f_j^*\|_T^2 \leqslant C\frac{s_j}{\vartheta^2}\frac{\xi}{\sqrt{mT}}\log^4(\xi d T), \tag{4.5}$$

with probability at least

$1 - \frac{1}{T} - \left(c_2 exp(-c_3(\xi + \log d)) + T^{-\left(\frac{1-c_0}{c_0}\right)}\right)$, *where* $m = T^{\frac{c_0 r_\beta - 1}{c_0 r_\beta}}$ *for β-mixing when $r_\beta \geqslant 1/c_0$, c_2 and c_3 are constants.*

Next, we present a result for the RKHS with infinitely many eigenvalues, but whose eigenvalues decay at a rate $\mu_\ell = (1/\ell)^{2\alpha}$ for some parameter $\alpha \geqslant 1/2$. Among other examples, this includes Sobolev spaces, say consisting of functions with α derivatives (e.g., Birman and Solomyak (1967); Gu (2013)).

Lemma 4.7. *For a univariate kernel with eigenvalue decay $\mu_\ell = (1/\ell)^{2\alpha}$ for some $\alpha \geqslant 1/2$, we have that $\tilde{\epsilon}_m = O\left(\left(\frac{\log^2(dT)}{\sqrt{m}}\right)^{\frac{2\alpha}{2\alpha+1}}\right)$.*

Corollary 4.8. *Under the same conditions as Theorem 4.3, consider a univariate kernel with eigenvalue decay* $\mu_\ell = (1/\ell)^{2\alpha}$ *for some* $\alpha \geqslant 1/2$. *Then there exists a constant* C *such that for each* $1 \leqslant j \leqslant d$,

$$\|\hat{f}_j - f_j^*\|_T^2 \leqslant C \frac{s_j}{\vartheta^2} \frac{\log^{\frac{8\alpha}{2\alpha+1}}(dT)}{\sqrt{m^{\frac{2\alpha-1}{2\alpha+1}} T}}, \tag{4.6}$$

with probability at least $1 - \frac{1}{T} - T^{-\left(\frac{1-c_0}{c_0}\right)}$, *where* $m = T^{\frac{c_0 r_\beta - 1}{c_0 r_\beta}}$ *for* β*-mixing when* $r_\beta \geqslant 1/c_0$.

Note that if $m = O(T)$, we obtain the rate $O\left(s_j T^{-\frac{2\alpha}{2\alpha+1}}\right)$ up to log factors which is optimal in the independent case.

4.5 Proof for the main result (Theorem 4.3)

At a high level, the proof for Theorem 4.3 follows similar steps to the proof of Theorem 1 in Raskutti et al. (2012). However a number of additional challenges arise when dealing with dependent data. The key challenge in the proof is that the traditional results for Rademacher complexities of RKHSs and empirical processes typically assume independence. These problems are addressed by Theorems 4.9 and 4.10 to follow which provide upper bounds for dependent empirical processes. Also note that previous techniques in Hall et al. (2016) are not applicable here because they require parametric assumptions which are amenable to analysis for high-dimensional parametric problems. In particular for the proof of Theorem 4.10, the common symmetrization technique fails to reduce the difference between expectations in $L^2(\mathbb{P}_T)$ and $L^2(\mathbb{P})$ to Rademacher complexity and martingale concentration inequality fails because of the non-linear transformation on the design matrix. Hence we use mixing assumptions to address both of these issues. Unlike previous works using mixing that only guarantee central limit theory, we quantify the convergence rate which then enables us to derive the upper bound on mean-squared error with high probability in the high-dimension setting.

Establishing the basic inequality

Our goal is to estimate the accuracy of $f_j^*(\cdot)$ for every integer j with $1 \leqslant j \leqslant d$. We denote the expected $\mathcal{L}_2(\mathbb{P})$ norm of a function g as $\|g\|_2^2 = \mathbb{E}\|g\|_T^2$ where the expectation is taken over the distribution of $(X_t)_{t=0}^T$. We begin the proof by establishing a basic inequality on the error function $\Delta_j(.) = \hat{f}_j(.) - f_j^*(.)$. Since $\hat{f}_j(.)$ and f_j^* are, respectively, optimal and feasible for (4.2), we are guaranteed that

$$\begin{aligned} &L_1(\hat{f}_j(X_t)) + \lambda_T \|\hat{f}_j\|_{T,1} + \lambda_H \|\hat{f}_j\|_{H,1} \\ &\leqslant L_1(f_j^*(X_t)) + \lambda_T \|f_j^*\|_{T,1} + \lambda_H \|f_j^*\|_{H,1}. \end{aligned}$$

Using our definition $w_{t,j} = \frac{1}{2}(\varphi(X_{t+1,j}) - \mathbb{E}[\varphi(X_{t+1,j})|X_t]) = \frac{1}{2}(\varphi(X_{t+1,j}) - Z'(v_j + f_j^*(X_t)))$ and recall that $L_1(f_j)$ is defined as

$$\frac{1}{2T}\sum_{t=0}^{T}(Z(v_j + f_j(X_t)) - (v_j + f_j(X_t))\varphi(X_{t+1,j}))$$

that inequality is the same as

$$\begin{aligned} &\frac{1}{2T}\sum_{t=1}^{T}(Z(v_j + \hat{f}_j(X_t)) - \hat{f}_j(X_t)(Z'(v_j + f_j^*(X_t)) + 2w_{t,j})) \\ &\quad + \lambda_T \|\hat{f}_j\|_{T,1} + \lambda_H \|\hat{f}_j\|_{H,1} \\ &\leqslant \frac{1}{2T}\sum_{t=1}^{T}(Z(v_j + f_j^*(X_t)) - f_j^*(X_t)(Z'(v_j + f_j^*(X_t)) + 2w_{t,j})) \\ &\quad + \lambda_T \|f_j^*\|_{T,1} + \lambda_H \|f_j^*\|_{H,1}. \end{aligned}$$

Let $B_Z(\cdot\|\cdot)$ denote the Bregman divergence induced by the strictly convex function Z, some simple algebra yields that

$$\begin{aligned}&\frac{1}{2T}\sum_{t=1}^{T} B_Z(\nu_j + \hat{f}_j(X_t)\|\nu_j + f_j^*(X_t)) \\ &\leqslant \frac{1}{T}\sum_{t=1}^{T}\Delta_j(X_t)w_{t,j} + \lambda_T\|\Delta_j\|_{T,1} + \lambda_H\|\Delta_j\|_{H,1}\end{aligned} \tag{4.7}$$

which we refer to as our basic inequality (see e.g. Geer (2000) for more details on the basic inequality).

Controlling the noise term

Let $\Delta_{j,k}(\cdot) = \hat{f}_{j,k}(\cdot) - f_{j,k}^*(\cdot)$ for any $k = 1, 2, ..., d$. Next, we provide control for the right-hand side of inequality (4.7) by bounding the Rademacher complexity for the univariate functions in terms of their $L^2(\mathbb{P}_T)$ and $\mathcal{H}$ norms. We point out that tools required for such control are not well-established in the dependent case which means that we first establish the Rademacher complexity result (Theorem 4.9) and the uniform convergence rate for averages in the empirical process (Theorem 4.10) for the dependent case (results for the independent case are provided as Lemma 7 in Raskutti et al. (2012)).

Theorem 4.9 (Rademacher complexity)**.** *Under Assumption 1, define the event*

$$\begin{aligned}\mathcal{A}_{m,T} = \Bigg\{&\forall (j,k) \in \{1,2,...,d\}^2, \forall \sigma \geqslant \tilde{\epsilon}_m, \\ &\sup_{\|f_{j,k}\|_{\mathcal{H}}\leqslant 1, \|f_{j,k}\|_2 \leqslant \sigma}\left|\frac{1}{T}\sum_{t=1}^{T} f_{j,k}(X_t)w_{t,j}\right| \leqslant \sqrt{2}\sqrt{\frac{m}{T}}\sigma^2\Bigg\}.\end{aligned}$$

Then $\mathbb{P}(\mathcal{A}_{m,T}) \geqslant 1 - \frac{1}{T}$.

Remark. We have a correction term $\sqrt{\frac{T}{m}}$ for $m < T$, in order to connect our Rademacher complexity result with mixing conditions. In the independent case,

$m = T$ which has been proven in prior work.

Theorem 4.10. *Define the event*

$$\mathcal{B}_{m,T} = \left\{ \sup_{j,k} \sup_{f_{j,k} \in B_H(1), \|f_{j,k}\|_2 \leqslant \gamma_m} |\|f_{j,k}\|_T - \|f_{j,k}\|_2| \leqslant \frac{\gamma_m}{2} \right\} \tag{4.8}$$

where $m = T^{\frac{c_0 r_\beta - 1}{c_0 r_\beta}}$ *for* β*-mixing with* $r_\beta \geqslant 1/c_0$*. Then* $\mathbb{P}(\mathcal{B}_{m,T}) \geqslant 1 - c_2 exp(-c_3 m \gamma_m^2) - T^{-\left(\frac{1-c_0}{c_0}\right)}$ *for some constants* c_2 *and* c_3*. Moreover, on the event* $\mathcal{B}_{m,T}$*, for any* $g \in B_{\mathcal{H}}(1)$ *with* $\|g\|_2 \geqslant \gamma_m$*,*

$$\frac{\|g\|_2}{2} \leqslant \|g\|_T \leqslant \frac{3}{2}\|g\|_2. \tag{4.9}$$

The proofs for Theorems 4.9 and 4.10 are provided in the appendix. Using Theorems 4.9 and 4.10, we are able to provide an upper bound on the noise term $\frac{1}{T}\sum_{t=1}^{T} \Delta_j(X_t) w_{t,j}$ in (4.7). In particular, recalling that $\tilde{\gamma}_m = c_1 \max\left\{ \epsilon_m, \tilde{\epsilon}_m, \sqrt{\frac{\log(dT)}{m}} \right\}$, we have the following lemma.

Lemma 4.11. *Given* $\tilde{\gamma}_m = \max(\gamma_m, \tilde{\epsilon}_m)$*, on the event* $\mathcal{A}_{m,T} \cap \mathcal{B}_{m,T}$*, we have:*

$$|\frac{1}{T}\sum_{t=1}^{T} f_{j,k}(X_t) w_{t,j}| \leqslant 4\sqrt{2}\sqrt{\frac{m}{T}}(\tilde{\gamma}_m \|f_{j,k}\|_T + \tilde{\gamma}_m^2 \|f_{j,k}\|_{\mathcal{H}}) \tag{4.10}$$

for any $f_{j,k} \in \mathcal{H}$*, for all* $(j,k) \in \{1, 2, ..., d\}^2$*.*

Exploiting decomposability

The reminder of our analysis involves conditioning on the event $\mathcal{A}_{m,T} \cap \mathcal{B}_{m,T}$. Recalling the basic inequality (4.7) and using Lemma 4.11, on the event $\mathcal{A}_{m,T} \cap \mathcal{B}_{m,T}$

defined in Theorems 4.9 and 4.10, we have:

$$\begin{aligned}
&\frac{1}{2T}\sum_{t=1}^{T} B_Z(\nu_j + \hat{f}_j(X_t)\|\nu_j + f_j^*(X_t)) \\
&\leqslant 4\sqrt{2}\sqrt{\frac{m}{T}}\tilde{\gamma}_m\|\Delta_j\|_{T,1} + 4\sqrt{2}\sqrt{\frac{m}{T}}\tilde{\gamma}_m^2\|\Delta_j\|_{\mathcal{H},1} \\
&\quad + \lambda_T\|\Delta_j\|_{T,1} + \lambda_H\|\Delta_j\|_{\mathcal{H},1}.
\end{aligned}$$

Recalling that S_j denotes the true support of the unknown function f_j^*, we define $\Delta_{j,S_j} := \sum_{k\in S_j}\Delta_{j,k}$, with a similar definition for Δ_{j,S_j^c}. We have that $\|\Delta_j\|_{T,1} = \|\Delta_{j,S_j}\|_{T,1} + \|\Delta_{j,S_j^c}\|_{T,1}$ with a similar decomposition for $\|\Delta_j\|_{\mathcal{H},1}$. We are able to show that conditioned on event $\mathcal{A}_{m,T}\cap\mathcal{B}_{m,T}$, the quantities $\|\Delta_j\|_{\mathcal{H},1}$ and $\|\Delta_j\|_{T,1}$ are not significantly larger than the corresponding norms as applied to the function Δ_{j,S_j}. First, notice that we can obtain a sharper inequality in the process of getting our basic inequality (4.7), that is,

$$\begin{aligned}
&\frac{1}{2T}\sum_{t=1}^{T} B_Z(\nu_j + \hat{f}_j(X_t)\|\nu_j + f_j^*(X_t)) \\
&\leqslant \frac{1}{T}\sum_{t=1}^{T}\Delta_j(X_t)w_{t,j} + \lambda_T(\|f_j^*\|_{T,1} - \|f_j^* + \Delta_j\|_{T,1}) \\
&\quad + \lambda_H(\|f_j^*\|_{\mathcal{H},1} - \|f_j^* + \Delta_j\|_{\mathcal{H},1}).
\end{aligned}$$

Using Lemma 4.11 and the fact that Bregman divergence is non-negative, on event $\mathcal{A}_{m,T}\cap\mathcal{B}_{m,T}$ we obtain

$$\begin{aligned}
0 \leqslant &4\sqrt{2}\sqrt{\frac{m}{T}}\tilde{\gamma}_m\|\Delta_j\|_{T,1} + \lambda_T(\|f_j^*\|_{T,1} - \|f_j^* + \Delta_j\|_{T,1}) \\
&+4\sqrt{2}\sqrt{\frac{m}{T}}\tilde{\gamma}_m^2\|\Delta_j\|_{\mathcal{H},1} + \lambda_H(\|f_j^*\|_{\mathcal{H},1} - \|f_j^* + \Delta_j\|_{\mathcal{H},1}).
\end{aligned}$$

Recall our choice $\lambda_T \geqslant 8\sqrt{2}\sqrt{\frac{m}{T}}\tilde{\gamma}_m$, $\lambda_H \geqslant 8\sqrt{2}\sqrt{\frac{m}{T}}\tilde{\gamma}_m^2$, that yields

$$\begin{aligned} 0 \leqslant & \frac{\lambda_T}{2}\|\Delta_j\|_{T,1} + \lambda_T(\|f_j^*\|_{T,1} - \|f_j^* + \Delta_j\|_{T,1}) \\ & + \frac{\lambda_H}{2}\|\Delta_j\|_{\mathcal{H},1} + \lambda_H(\|f_j^*\|_{\mathcal{H},1} - \|f_j^* + \Delta_j\|_{\mathcal{H},1}). \end{aligned}$$

Now, for any $k \in S_j^C$, we have

$$\begin{aligned} \|f_{j,k}^*\|_T - \|f_{j,k}^* + \Delta_{j,k}\|_T = -\|\Delta_{j,k}\|_T, \\ \text{and } \|f_{j,k}^*\|_{\mathcal{H}} - \|f_{j,k}^* + \Delta_{j,k}\|_{\mathcal{H}} = -\|\Delta_{j,k}\|_{\mathcal{H}}. \end{aligned}$$

On the other hand, for any $k \in S_j$, the triangle inequality yields

$$\|f_{j,k}^*\|_T - \|f_{j,k}^* + \Delta_{j,k}\|_T \leqslant \|\Delta_{j,k}\|_T$$

with a similar inequality for the terms involving $\|\cdot\|_{\mathcal{H}}$. Given those bounds, we conclude that

$$\begin{aligned} 0 \leqslant & \frac{\lambda_T}{2}\|\Delta_j\|_{T,1} + \lambda_T(\|\Delta_{j,S_j}\|_{T,1} - \|\Delta_{j,S_j^C}\|_{T,1}) \\ & + \frac{\lambda_H}{2}\|\Delta_j\|_{\mathcal{H},1} + \lambda_H(\|\Delta_{j,S_j}\|_{\mathcal{H},1} - \|\Delta_{j,S_j^C}\|_{\mathcal{H},1}). \end{aligned} \tag{4.11}$$

Using the triangle inequality $\|\Delta_j\| \leqslant \|\Delta_{j,S_j}\| + \|\Delta_{j,S_j^C}\|$ for any norm and rearranging terms, we obtain

$$\|\Delta_{j,S_j^C}\|_{T,1} + \|\Delta_{j,S_j^C}\|_{\mathcal{H},1} \leqslant 3(\|\Delta_{j,S_j}\|_{T,1} + \|\Delta_{j,S_j}\|_{\mathcal{H},1}),$$

which implies

$$\|\Delta_j\|_{T,1} + \|\Delta_j\|_{\mathcal{H},1} \leqslant 4(\|\Delta_{j,S_j}\|_{T,1} + \|\Delta_{j,S_j}\|_{\mathcal{H},1}). \tag{4.12}$$

This bound allows us to exploit the sparsity assumption, since in conjunction with Lemma 4.11, we have now bounded the right-hand side of the basic inequality (4.7) in terms involving only Δ_{j,S_j}. In particular, still conditioning on event $\mathcal{A}_{m,T} \cap \mathcal{B}_{m,T}$

and applying (4.12), we obtain

$$\begin{aligned}&\frac{1}{2T}\sum_{t=1}^{T} B_Z(\nu_j + \hat{f}_j(X_t)\|\nu_j + f_j^*(X_t))\\&\leqslant C\sqrt{\frac{m}{T}}\{\tilde{\gamma}_m\|\Delta_{j,S_j}\|_{T,1} + \tilde{\gamma}_m^2\|\Delta_{j,S_j}\|_{\mathcal{H},1}\},\end{aligned}$$

for some constant C, where we have recalled our choices $\lambda_T = O(\sqrt{\frac{m}{T}}\tilde{\gamma}_m)$ and $\lambda_H = O(\sqrt{\frac{m}{T}}\tilde{\gamma}_m^2)$. Finally, since both $\hat{f}_{j,k}$ and $f_{j,k}^*$ belong to $B_{\mathcal{H}}(1)$, we have

$$\|\Delta_{j,k}\|_{\mathcal{H}} \leqslant \|\hat{f}_{j,k}\|_{\mathcal{H}} + \|f_{j,k}^*\|_{\mathcal{H}} \leqslant 2,$$

which implies that $\|\Delta_{j,S_j}\|_{\mathcal{H},1} \leqslant 2s_j$, and hence

$$\begin{aligned}&\frac{1}{2T}\sum_{t=1}^{T} B_Z(\nu_j + \hat{f}_j(X_t)\|\nu_j + f_j^*(X_t))\\&\leqslant C\sqrt{\frac{m}{T}}(\tilde{\gamma}_m\|\Delta_{j,S_j}\|_{T,1} + s_j\tilde{\gamma}_m^2).\end{aligned}$$

Exploiting strong convexity

On the other hand, we are able to bound the Bregman divergence term on the left-hand side as well by noticing that (4.12) implies

$$\|\Delta_j\|_{\mathcal{H},1} \leqslant 16s_j, \tag{4.13}$$

since $\hat{f}_{j,k}$ and $f^*_{j,k}$ belong to $B_{\mathcal{H}}(1)$ with $\|\hat{f}_{j,k}\|_\infty \leqslant 1$ and $\|f^*_{j,k}\|_\infty \leqslant 1$. Using bound (4.13), for any t, we conclude that

$$\begin{aligned}
|\hat{f}_j(X_t)| &= |\Delta_j(X_t) + f^*_j(X_t)| \\
&= |\sum_{k=1}^{d} \Delta_{j,k}(X_{t,k}) + f^*_j(X_t)| \\
&\leqslant \sum_{k=1}^{d} \|\Delta_{j,k}\|_{\mathcal{H}} \max_k \sqrt{\mathcal{K}(X_{t,k}, X_{t,k})} + |f^*_j(X_t)| \\
&\leqslant 16\sqrt{\sum_{i=1}^{\infty} \mu_i s_j} + |f^*_j(X_t)| \\
&\leqslant \left(16\sqrt{\sum_{i=1}^{\infty} \mu_i} + 1\right) s_{\max}.
\end{aligned}$$

Therefore, $\nu_j + \hat{f}_j(X_t), \nu_j + f^*_j(X_t) \in [\nu_{\min} - (16\sqrt{\sum_{i=1}^{\infty} \mu_i} + 1)s_{\max}, \nu_{\max} + (16\sqrt{\sum_{i=1}^{\infty} \mu_i} + 1s_{\max}]$ where we have function $Z(\cdot)$ is ϑ-strongly convex given Assumption 2. Hence

$$\frac{\vartheta}{2}\|\Delta_j\|_T \leqslant C\sqrt{\frac{m}{T}}\{\tilde{\gamma}_m\|\Delta_{j,S_j}\|_{T,1} + s_j\tilde{\gamma}_m^2\}. \tag{4.14}$$

Relating the $\mathcal{L}^2(\mathbb{P}_T)$ and $\mathcal{L}^2(\mathbb{P})$ norms

It remains to control the term $\|\Delta_{j,S_j}\|_{T,1} = \sum_{k\in S_j} \|\Delta_{j,k}\|_T$. Ideally we would like to upper bound it by $\sqrt{s_j}\|\Delta_{j,S_j}\|_T$. Such an upper bound would follow immediately if it were phrased in terms of the $\|\cdot\|_2$ rather than the $\|\cdot\|_T$ norm, but there are additional cross-terms with the empirical norm. Accordingly, we make use of two lemmas that relate the $\|\cdot\|_T$ norm and the population $\|\cdot\|_2$ norms for functions in $\mathcal{F}_j := \cup_{S_j\subset\{1,2,\ldots,d\}|S_j|=s_j} H_j(S_j)$.

In the statements of these results, we adopt the notation g_j and $g_{j,k}$ (as opposed to f_j and $f_{j,k}$) to be clear that our results apply to any $g_j \in \mathcal{F}_j$. We first provide an upper bound on the empirical norm $\|g_{j,k}\|_T$ in terms of the associated $\|g_{j,k}\|_2$ norm,

one that holds uniformly over all components $k = 1, 2, ..., d$.

Lemma 4.12. *On event* $\mathcal{B}_{m,T}$,

$$\|g_{j,k}\|_T \leqslant 2\|g_{j,k}\|_2 + \gamma_m, \text{ for all } g_{j,k} \in B_{\mathcal{H}}(2), \tag{4.15}$$

for any $(j, k) \in \{1, 2, ..., d\}^2$.

We now define the function class $2\mathcal{F}_j := \{f + f' | f, f' \in \mathcal{F}_j\}$. Our second lemma guarantees that the empirical norm $\|\cdot\|_T$ of any function in $2\mathcal{F}_j$ is uniformly lower bounded by the norm $\|\cdot\|_2$.

Lemma 4.13. *Given properties of* γ_m *and* $\delta_{m,j}^2 = c_4\{\frac{s_j \log d}{m} + s_j\epsilon_m^2\}$, *we define the event*

$$\begin{aligned}\mathcal{D}_{m,T} = \{\forall j \in [1, 2, ..., d], \|g_j\|_T \geqslant \|g_j\|_2/2 \\ \text{for all } g_j \in 2\mathcal{F}_j \text{ with } \|g_j\|_2 \geqslant \delta_{m,j}\}\end{aligned} \tag{4.16}$$

where $m = T^{\frac{c_0 r_\beta - 1}{c_0 r_\beta}}$ *for* β*-mixing with* $r_\beta \geqslant 1/c_0$. *Then we have* $\mathbb{P}(\mathcal{D}_{m,T}) \geqslant 1 - c_2 exp(-c_3 m(\min_j \delta_{m,j}^2)) - T^{-\left(\frac{1-c_0}{c_0}\right)}$ *where* c_2, c_3 *and* c_4 *are constants.*

Note that while both results require bounds on the univariate function classes, they do not require global boundedness assumptions-that is, on quantities of the form $\|\sum_{k \in S_j} g_{j,k}\|_\infty$. Typically, we expect that the $\|\cdot\|_\infty$-norms of functions $g_j \in \mathcal{F}_j$ scale with s_j.

Completing the proof

Using Lemmas 4.12 and 4.13, we complete the proof of the main theorem. For the reminder of the proof, let us condition on the events $\mathcal{A}_{m,T} \cap \mathcal{B}_{m,T} \cap \mathcal{D}_{m,T}$.

Conditioning on the event $\mathcal{B}_{m,T}$, we have

$$\begin{aligned} \|\Delta_{j,S_j}\|_{T,1} &= \sum_{k \in S_j} \|\Delta_{j,k}\|_T \\ &\leqslant 2 \sum_{k \in S_j} \|\Delta_{j,k}\|_2 + s_j \gamma_m \\ &\leqslant 2\sqrt{s_j} \|\Delta_{j,S_j}\|_2 + s_j \gamma_m. \end{aligned} \tag{4.17}$$

Our next step is to bound $\|\Delta_{j,S_j}\|_2$ in terms of $\|\Delta_{j,S_j}\|_T$ and $s_j \gamma_m$. We split our analysis into two cases.

Case 1: If $\|\Delta_{j,S_j}\|_2 < \delta_{m,j} = \Theta(\sqrt{s_j}\gamma_m)$, then we conclude that $\|\Delta_{j,S_j}\|_{1,T} \leqslant C s_j \gamma_m$.

Case 2: Otherwise, we have $\|\Delta_{j,S_j}\|_2 \geqslant \delta_{m,j}$. Note that the function $\Delta_{j,S_j} = \sum_{k \in S_j} \Delta_{j,k}$ belongs to the class $2\mathcal{F}_j$ so that it is covered by the event $\mathcal{D}_{m,T}$. In particular, conditioned on the event $\mathcal{D}_{m,T}$, we have $\|\Delta_{j,S_j}\|_2 \leqslant 2\|\Delta_{j,S_j}\|_T$. Combined with the previous bound (4.17), we conclude that

$$\|\Delta_{j,S_j}\|_{T,1} \leqslant C\{\sqrt{s_j}\|\Delta_{j,S_j}\|_{T,2} + s_j \gamma_m\}.$$

Therefore in either case, a bound of the form $\|\Delta_{j,S_j}\|_{T,1} \leqslant C\{\sqrt{s_j}\|\Delta_{j,S_j}\|_{T,2} + s_j \gamma_m\}$ holds. Substituting the inequality in the bound (4.14) yields

$$\frac{\vartheta}{2}\|\Delta_j\|_T^2 \leqslant C_1 \sqrt{\frac{m}{T}}(\sqrt{s_j}\tilde{\gamma}_m \|\Delta_{j,S_j}\|_T + s_j \tilde{\gamma}_m^2).$$

The term $\|\Delta_{j,S_j}\|_T$ on the right side of the inequality is bounded by $\|\Delta_j\|_T$ and the inequality still holds after replacing $\|\Delta_{j,S_j}\|_T$ by $\|\Delta_j\|_T$. Through rearranging terms in that inequality, we get,

$$\|\Delta_j\|_T^2 \leqslant 2C_1 \frac{1}{\vartheta}\sqrt{\frac{m}{T}}\left(\sqrt{s_j}\tilde{\gamma}_m \|\Delta_j\|_T + s_j \tilde{\gamma}_m^2\right). \tag{4.18}$$

Because $\frac{m}{T} \leqslant 1$ and $\frac{1}{\vartheta} \geqslant 1$, we can relax the inequality to

$$\begin{aligned} &\|\Delta_j\|_T^2 \leqslant \\ &2C_1 \left(\frac{1}{\vartheta} \left(\frac{m}{T} \right)^{1/4} \sqrt{s_j} \tilde{\gamma}_m \|\Delta_j\|_T + \frac{1}{\vartheta^2} \left(\frac{m}{T} \right)^{1/2} s_j \tilde{\gamma}_m^2 \right). \end{aligned} \tag{4.19}$$

We can derive a bound on $\|\Delta_j\|_T$ from that inequality, which is

$$\begin{aligned} \|\Delta_j\|_T^2 &\leqslant C_2 \frac{s_j}{\vartheta^2} \sqrt{\frac{m}{T}} \tilde{\gamma}_m^2 \\ &= C_2 \frac{s_j}{\vartheta^2} \sqrt{\frac{m}{T}} \left(\frac{\log(dT)}{m} + \max(\epsilon_m, \tilde{\epsilon}_m)^2 \right) \\ &= C_2 \frac{s_j}{\vartheta^2} \left(\frac{\log(dT)}{\sqrt{mT}} + \sqrt{\frac{m}{T}} \max(\epsilon_m, \tilde{\epsilon}_m)^2 \right) \\ &= C_2 \frac{s_j}{\vartheta^2} \left(\frac{\log(dT)}{\sqrt{mT}} + \sqrt{\frac{m}{T}} \tilde{\epsilon}_m^2 \right), \end{aligned} \tag{4.20}$$

where C_2 only depends on C_1. That completes the proof.

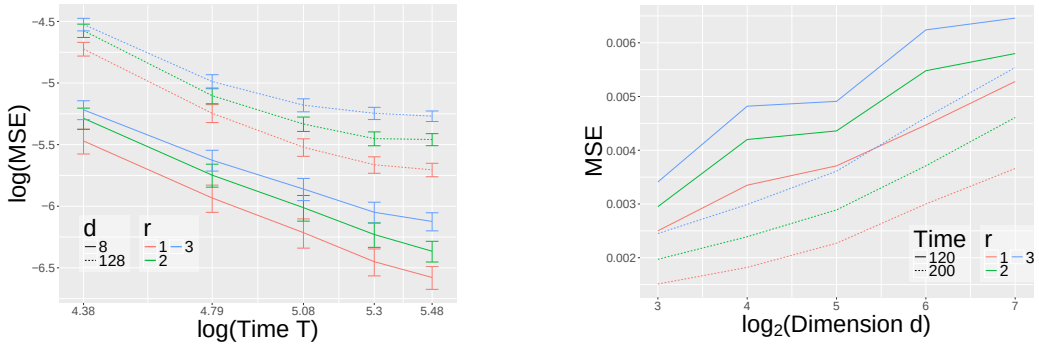
4.6 Numerical experiments

Our experiments are two-fold. First we perform simulations that validate the theoretical results in Section 4.4. We then apply the SpAM framework on a Chicago crime dataset and show its improvement in prediction error and ability to discover additional interesting patterns beyond the parametric model. Instead of using the sparse and smooth objective in this Chapter, we implement a computationally faster approach through the R CRAN package "SAM", which includes the first penalty term $\|f_j\|_{1,T}$ but not the second term $\|f_j\|_{1,\mathcal{H}}$ (Zhao and Liu (2012)). We also implemented our original optimization problem in 'cvx' however this approach does not scale. Hence we use the "SAM" package.

Simulations

We validate our theoretical results with experimental results performed on synthetic data. We generate many trials with known underlying parameters and then compare the estimated function values with the true values. For all trials the constant offset vector ν is set identically at 0. Given an initial vector X_0, samples are generated consecutively using the equation $X_{t+1,j} = f_j^*(X_t) + w_{t+1,j}$, where $w_{t+1,j}$ is the noise chosen from a uniform distribution on the interval $[-0.4, 0.4]$ and f_j^* is the signal function, which means that the log-partition function $Z(\cdot)$ is the standard quadratic $Z(x) = \frac{1}{2}x^2$ and the sufficient statistic $\varphi(x) = x$. The signal function f_j^* is assigned in two steps to ensure that the Markov chain mixes and we incorporate sparsity. In the first step, we define sparsity parameters $\{s_j\}_{j=1}^d$ all to be 3 (for convenience) and set up a d by d sparse matrix A^*, which has 3 non-zero off-diagonal values on each row drawn from a uniform distribution on the interval $[-\frac{1}{2s}, \frac{1}{2s}]$ and all 1 on diagonals. In the second step, given a polynomial order parameter r, we map each value $X_{t,k}$ in vector X_t to $(\Phi_1(X_{t,k}), \Phi_2(X_{t,k}), ..., \Phi_r(X_{t,k}))$ in $\mathbb{R}^r$ space, where $\Phi_i(x) = \frac{x^i}{i!}$ for any i in $\{1, 2, .., r\}$. We then randomly generate standardized vectors $(b_{j,k}^1, b_{j,k}^2, b_{j,k}^3)$ for every (j, k) in $\{1, 2, ..., d\}^2$ and define f_j^* as $f_j^*(X_t) = \sum_{k=1}^d A_{j,k}^*(\sum_{i=1}^r b_{j,k}^i \Phi_i(X_{t,k}))$. The tuning parameter λ_T is chosen to be $3\sqrt{\log(dr)/T}$ following the theory. We focus on polynomial kernels for which we have theoretical guarantees in Lemma 4.5 and Corollary 4.6 since the "SAM" package is suited to polynomial basis functions.

The simulation is repeated 100 times with 5 different values of d ($d = 8, 16, 32, 64, 128$), 5 different numbers of time points ($T = 80, 120, 160, 200, 240$), and 3 different polynomial order parameters ($r = 1, 2, 3$) for each repetition. These design choices are made to ensure the sequence $(X_t)_{t=0}^T$ is stable and mixes. Other experimental settings were also run with similar results. We present the mean squared error (MSE) of our estimates in Fig. 4.1. Since we select r values from the same vector $(b_{j,k}^1, b_{j,k}^2, b_{j,k}^3)$ for all polynomial order parameters, the MSE for different r is comparable and will be higher for larger r because of stronger absolute signal value. In Fig. 4.1(a), we see that MSE decreases in the rate between T^{-1} and $T^{-0.5}$ for all

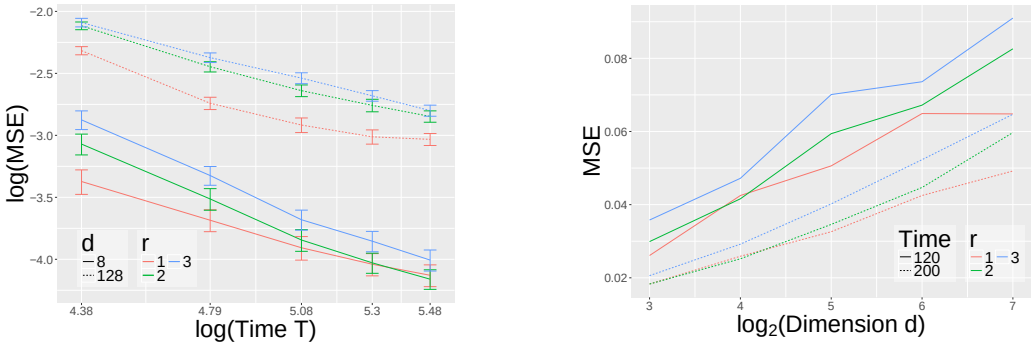


(a) log(MSE) vs. log(Time T) (b) MSE vs. log(Dimension d)

Figure 4.1: (a) shows the logarithm of MSE over a range of log T values, from 80 to 240 under the regression setting. (b) shows the MSE over a range of log d values, from 8 to 128 under the regression setting. In all plots the mean value of 100 trials is shown, with error bars denoting the 90% confidence interval for plot (a). For plot (b), we also have error bars results but we do not show them for the cleanness of the plot.

combinations of r and d. For larger d, MSE is larger and the rate becomes slower. In Fig. 4.1(b), we see that MSE increases slightly faster than the log d rate for all combinations of r and T which is consistent with Theorem 4.3 and Corollary 1.

Similarly we consider the Poisson link function and Poisson process for modeling count data. Given an initial vector X_0, samples are generated consecutively using the equation $X_{t+1,j} \sim \text{Poisson}(\exp(f_j^*(X_t)))$, where f_j^* is the signal function. The signal function f_j^* is again assigned in two steps to ensure the Poisson Markov process mixes. In the first step, we define sparsity parameters $\{s_j\}_{j=1}^d$ all to be 3 and set up a d by d sparse matrix A^*, which has 3 non-zero values on each row set to be -2 (this choice ensures the process mixes). In the second step given a polynomial order parameter r, we map each value $X_{t,k}$ in vector X_t to $(\Phi_1(X_{t,k}), \Phi_2(X_{t,k}), ..., \Phi_r(X_{t,k}))$ in $\mathbb{R}^r$, where $\Phi_i(x) = \frac{x^i}{i!}$ for any i in $\{1, 2, .., r\}$. We then randomly generate standardized vectors $(b_{j,k}^1, b_{j,k}^2, b_{j,k}^3)$ for every (j, k) in $\{1, 2, ..., d\}^2$ and define f_j^* as $f_j^*(X_t) = \sum_{k=1}^d A_{j,k}^* (\sum_{i=1}^r b_{j,k}^i \Phi_i(X_{t,k}))$. The tuning parameter λ_T is chosen to be $1.3(\log d \log T)(\sqrt{r}/\sqrt{T})$. The simulation is repeated



(a) log(MSE) vs. log(Time T)

(b) MSE vs. log(Dimension d)

Figure 4.2: (a) shows the logarithm of MSE behavior over a range of log T values, from 80 to 240 for the Poisson process. (b) shows the MSE behavior over a range of log d values, from 8 to 128 for the Poisson process. In all plots the mean value of 100 trials is shown, with error bars denoting the 90% confidence interval for plot (a). For plot (b), we also have error bars results but we do not show them for the cleanness of the plot.

100 times with 5 different numbers of time series ($d = 8, 16, 32, 64, 128$), 5 different numbers of time points ($T = 80, 120, 160, 200, 240$) and 3 different polynomial order parameters ($r = 1, 2, 3$) for each repetition. These design choices are made to ensure the sequence $(X_t)_{t=0}^{T}$ mixes. Other experimental settings were also considered with similar results, but are not included due to space constraints.

We present the mean squared error (MSE) of our estimations in Fig. 4.2. Since we select r values from the same vector $(b^1_{j,k}, b^2_{j,k}, b^3_{j,k})$ for all polynomial order parameters, the MSE tends to be higher for larger r because the process has larger variance. In Fig. 4.2 (a), we see that MSE decreases in the rate between T^{-1} and $T^{-0.5}$ for all combinations of r and d. For larger d, MSE is larger and the rate becomes slower. In Fig. 4.2 (b), we see that MSE increases slightly faster than the $\log(d)$ rate for all combinations of r and T which is consistent with our theory.

Chicago crime data

We now evaluate the performance of the SpAM framework on a Chicago crime dataset to model incidents of severe crime in different community areas of Chicago. [1]. We are interested in predicting the number of homicide and battery (severe crime) events every two days for 76 community areas over a two month period. The recorded time period is April 15, 2012 to April 14, 2014 as our training set and we choose the data from April 15, 2014 to June 14, 2014 to be our test data. In other words, we consider dimension $d = 76$ and time range $T = 365$ for training set and $T = 30$ for the test set. Though the dataset has records from 2001, we do not use all previous data to be our training set since we do not have stationarity over a longer period. We choose a 2 month test set for the same reason. We choose time horizon to be two days so that number of crimes is counted over each two days. Since we are modeling counts, we use the Poisson GLM and the exponential link $Z(x) = e^x$.

We apply the "SAM" package for this task using B-spline as our basis. The degrees of freedom r are set to $1, 2, 3$ or 4, where 1 means that we only use linear basis. In the first part of the experiment, we choose the tuning parameter λ_T using 3-cross validation; the validation pairs are chosen as 60 days back (i.e., February 15, 2012 to February 14, 2014 as the training set and February 15, 2014 to April 14, 2014 as the testing set), 120 days back and 180 days back from April 15, 2012 and April 15, 2014 but with the same time range as the training set and test set. Then we test SpAM with this choice of λ_T. The performance of the model is measured by Pearson chi-square statistic, which is defined as

$$\frac{1}{30}\sum_{t=0}^{29}\frac{(X_{t+1,j}-\hat{f}_j(X_t))^2}{\hat{f}_j(X_t)}$$

on the 30 test points for the j^{th} community area. The Pearson chi-square statistic is

[1] This dataset reflects reported incidents of crime that occurred in the City of Chicago from 2001 to present. Data is extracted from the Chicago Police Department's CLEAR (Citizen Law Enforcement Analysis and Reporting) system `https://data.cityofchicago.org`

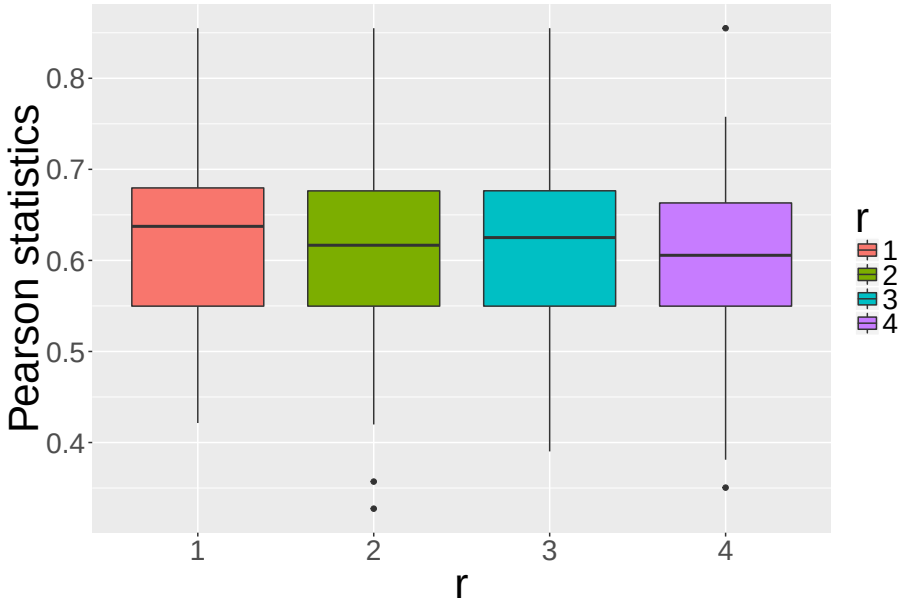
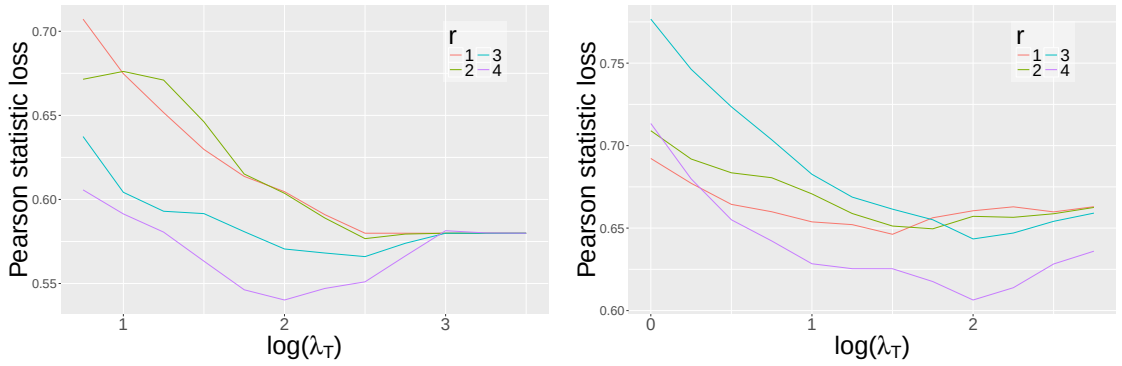


Figure 4.3: The boxplot shows the performance of SpAM on crime data measured by Pearson statistic for $r = 1, 2, 3, 4$-degrees of freedom in B-spline basis.

commonly used as the goodness-of-fit measure for discrete observations Hosmer et al. (1997). In Fig. 4.3, we show a box plot for the test loss on 17 non-trivial community areas, where "trivial" means that the number of crimes in the area follows a Poisson distribution with constant rate, which tells us that there is no relation between that area and other areas and no relation between different time. From Fig. 4.3, we can see that as basis become more complex from linear to B-spline with 4 degrees of freedom, the performance of fitting is gradually (although not majorly) improved. The main benefit of using higher-order (non-parametric) basis is revealed in Fig. 4.4 where we pick two community areas and plot the λ_T path performance for every r in Fig. 4.4. In the examples of two community areas shown in Fig. 4.4, we can see that the non-parametric SpAM has a lower test loss than linear model ($r = 1$). For community area 34, when r is set to be 3 and 4, the SpAM model discovers meaningful influences of other community areas on that area while the model with r equal to 1 or 2 choose a constant Poisson process as the best fitting. A similar conclusion holds for community area 56. Here $r = 1$ corresponds

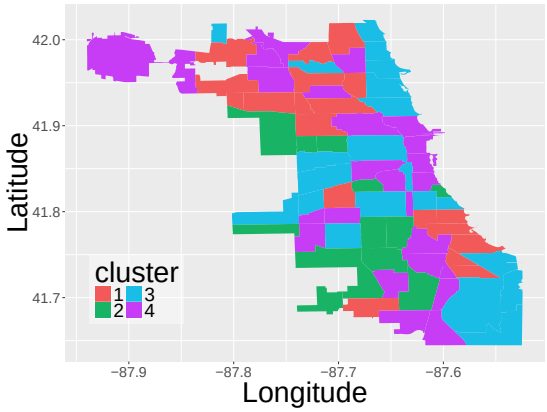


(a) Performance on community area 34 (b) Performance on community area 56

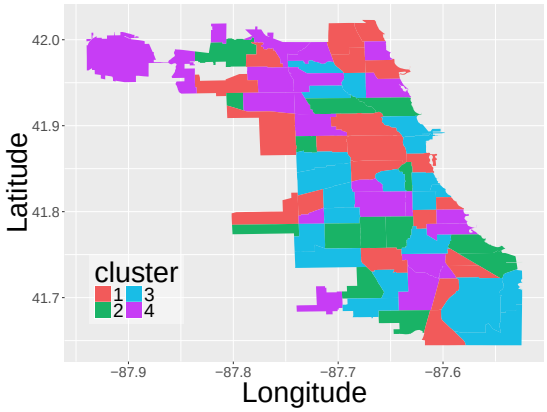
Figure 4.4: (a) shows the Pearson statistic loss on the number of crimes in community area 34. (b) shows the Pearson statistic loss on the number of crimes in community area 56.

to the parametric model in Hall et al. (2016).

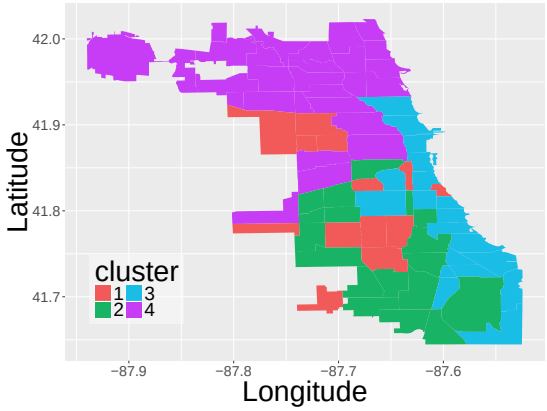
Finally, we present a visualization of the estimated network for the Chicago crime data. Since the estimated model is a network, the sparse patterns can be represented as an adjacency matrix where 1 in the i^{th} row and j^{th} column means that the i^{th} community area has influence on the j^{th} community area and 0 means no effect. Given the adjacency matrix, we can use spectral clustering to generate clusters for different polynomial order r's used in SpAM model, which are shown in Figs. 4.5 (a) and (b). For each case, even the location information is not used in learning at all, we find that the close community areas are clustered together. We see that the patterns from the non-parametric model ($r = 3$) is different from the parametric generalized linear model ($r = 1$) and they seem more smooth. It tells us that the non-parametric model proposed in this Chapter can help us to discover additional patterns beyond the linear model. Even in other tasks, the clusters cannot represent the location information very well. In Binkiewicz et al. (2017); Zhang et al. (2018), the authors proposed a covariate-assisted method to deal with this problem, which applies spectral clustering on $L + \lambda X^T X$, where L is the adjacency matrix, X are the covariates (latitude and longitude in our case), and λ is a tuning parameter.



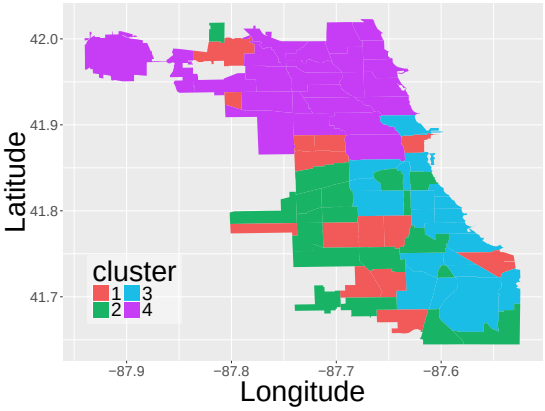
(a) $r = 3$



(b) $r = 1$



(c) $r = 3$ using location information



(d) $r = 1$ using location information

Figure 4.5: (a) shows the clusters given by spectral clustering using the adjacency matrix from SpAM with polynomial order $r = 3$, (b) shows the clusters when polynomial order $r = 1$. To derive clusters in (c), compared to (a), we add location information to the adjacency matrix for $r = 3$. Similarly, we get clusters in (d) using location information compared to (b) for $r = 1$.

By using location information as the assisted covariate in spectral clustering, we obtain results in Fig. 4.5 (c)(d). Since the location information is used, we see in both cases that community areas are almost clustered in four groups based on location information. Again, we find that the patterns from non-parametric model is different from the linear model and the separation between clusters is slightly clearer.

5 UNDERSTAND THE FLOW OF INFORMATION IN DEEP PROBABILISTIC MODELS: EXTEND STATISTICAL MODELS TO DEEP STRUCTURE

5.1 Introduction

In this Chapter, we study the case when data is huge and the model trained to fit it is complicated to understand. While we discussed in previous Chapters about how to combine multiple heterogeneous datasets, and analyze high-dimensional spatiotemporal data, we find that in the last decade, prediction tasks when sufficient data is available can be significantly facilitated by the use of deep neural networks. However, deep neural networks lack interpretability and probabilistic measurements, which make their predictions hard to explain and understand in scientific studies. In this Chapter, we show how kernel methods together with deep probabilistic models can help us to obtain deep neural networks with interpretability and uncertainty measurements. To complete that, we derive a framework to understand the flow of information in deep probabilistic models.

Deep probabilistic models are a core topic of interest within modern deep learning. Motivated by a spectrum of applications that benefit from formulations that roughly fall under the umbrella of deep probabilistic models – namely – Bayesian neural networks (BNNs), deep Gaussian processes (DGPs), variational autoencoders (VAEs) and others Hernández-Lobato and Adams (2015); Gal and Ghahramani (2016); Damianou and Lawrence (2013); Bui et al. (2016); Wilson et al. (2016), there is a growing interest in better understanding their properties, both from a theoretical as well as a practical perspective. Our goal in this Chapter is to study mechanisms to characterize the *information flow* for deep probabilistic models, i.e., the flow of representations from the input features to the output prediction passing through the units that make up the network. While this is motivated by and should facilitate applications where interpretability is important, in principle, a clear understanding of these properties would also allow a more rigorous analysis of the model and

evaluating or verifying if it has desirable statistical properties. For general deep neural networks (DNNs), there is a sizable (and growing) body of work that tackles information flow in a number of interesting ways. For example, one could construct procedures that seek to explain *individual* feature importance, for example by computing input gradients and decomposing predictions Shrikumar et al. (2017); Ancona et al. (2018); Sundararajan et al. (2017). Concurrently, there are also proposals based on attention-based models Xu et al. (2015); Mnih et al. (2014), which are able to localize which parts of the images critically drive the decision making process of a DNN model. Expectedly, this understanding will often, although not always, enable better interpretation of *why* the prediction was a certain class (and not another). More recent works have also sought to use simpler interpretable models to explain the prediction from a DNN Chen et al. (2018). Despite this evolving body of work, we find that analogous results for deep probabilistic models are either lacking or still in a nascent stage.

One should be able to, at least in principle, adapt existing information flow or semantic attribution strategies which work for general DNNs, to the deep probabilistic models setting. But we find that an overwhelming majority of such procedures natively focus on individual feature importance, e.g., pixel saliency maps for images. But inspecting coefficients of individual predictor variables is often insufficient in many classical statistical models – and we are often interested in statistical interactions and on occasion, higher order interactions. We find, as has been noted recently in Tsang et al. (2018) that extending available feature importance and sensitivity analysis methods for DNNs to this general case is non-trivial. It turns out, however, that if we use simple strategies, namely imposing assumptions on the structure of the function class in a manner we will describe shortly, this problem turns out to be a little more tractable. This strategy has precedence – for example, in classical statistical models, assuming an additive or a hierarchical structure is quite common for numerical reasons or to better analyze its statistical behavior Hastie and Tibshirani (1986); Gelman and Hill (2006); Huang et al. (2010). Therefore, it seems that an assumption, similar as above, i.e., an explicitly specified structure on the function class, may be a good idea for deep probabilistic models as well. If successful, this

will help characterize information flow, not merely back to individual features but will also help understand the relevance of higher order statistical interactions on the output. While not stated in this form, such an assumption, nonetheless, underlies capsule structures Sabour et al. (2017) and multi-resolution analysis (MRA) in CNNs Mallat (2016); Angles and Mallat (2018).

What is the structure assumption on the function class? Build function with a computation skeleton. To better understand and characterize the function class which our deep probabilistic model corresponds to, we will make use of the so-called "computation skeleton" idea in Daniely et al. (2016). In Daniely et al. (2016), the computation skeleton is used to study the relationship between DNNs and kernels and the effect of depth. Here, we find that the computation skeleton idea helps us capture the overall structure of the DNN and the function class via an easier-to-analyze "gadget". Importantly, it can be generalized in a way where we can evaluate the functions being learned (by the deep probabilistic model) via analyzing the corresponding computation skeleton (and the functionality we insert in the different parts of the skeleton). This significantly facilitates analyzing the flow of information because we now only need to analyze information flow from the input to the output *over* the computation skeleton.

Other advantages. As a by-product, the foregoing assumption also helps us to study the relationship between BNNs and DGPs, the effect of layer width and dataset shape in the deep probabilistic model setting. For instance, some recent papers describe designing deep probabilistic models that can be both understood as Bayesian neural networks and deep Gaussian processes Gal and Ghahramani (2016); Cutajar et al. (2017); Salimbeni and Deisenroth (2017). It turns out that our mechanisms to impose structure on the function class lead to an interesting theoretical perspective describing the relation between Bayesian neural networks and deep Gaussian processes through kernels. Such a result, helps us tie various deep probabilistic models together in our framework, which can easily adjust to different probabilistic formulations. Additionally, we could use this general framework to flexibly choose between different structures and even uncertainty estimation schemes. All these benefits essentially come for free – the framework

retains all useful empirical properties from BNNs such as mini-batch training, works on large-scale datasets and yields the expressive power of DGPs with kernels. Finally, if desired, one could also easily compare various deep probabilistic models using this framework as a tool.

Additional benefits for interpretability. With the computation skeleton framework for deep probabilistic models in hand, it will be easy to design specific structures to obtain interpretable results. Instead of studying feature importance, we show that the framework can lead to a solution to study statistical interactions, which is important for interpretability in biomedical and financial applications but has received scant attention Tsang et al. (2018). If an output depends on several features, one is often interested in changing some features to evaluate how it affects the response. In doing so, we must guarantee that other "uncontrolled" features do *not* influence the response. This confound is called *interaction*: the simultaneous influence of several features on the outcome is *not additive* and the features may jointly affect the outcome. Interpretability means understanding how predictors influence the outcome. But failing to detect statistical interactions causes problems in inferring the features' influence (e.g., the Simpson's paradox). A general DNN architecture permits *all* features to interact, without the ability to control for the nuisance terms. In statistics, we may use a fully additive statistical model with ANOVA decomposition. Similarly, we propose an additive structure on the network and apply post-training ANOVA decomposition to detect statistical interactions: what may be called a Bayesian *additive* neural network (BANN).

Our contributions.

i) We derive a framework to understand information flow for deep probabilistic models.

ii) We extend the "structure on function" assumption used in statistics to DNNs and deep probabilistic models, which helps us to obtain statistical measures beyond feature importance (namely, interactions) for interpretability.

iii) Our framework ties various deep probabilistic models together including BNNs, DGPs, DKL and others. The analysis helps us to understand the similarities/differences between various deep probabilistic models.

iv) We leverage our structure assumption explicitly – as an additive structure – to obtain Bayesian additive neural networks (BANN), which provide competitive results to DNNs but with other benefits.

Preliminaries

In this section, we briefly review deep Gaussian process and variational inference schemes to setup the rest of our presentation. We also review the classical Analysis of Variance (ANOVA) decomposition in statistics which we use later.

Gaussian processes (GP) and deep Gaussian processes (DGPs). Consider the inference task for a stochastic function $f : \mathbb{R}^p \to \mathbb{R}$, given a likelihood $p(y|f)$ and a set of n observations $\mathbf{y} = (y_1, ..., y_n)^\top \in \mathbb{R}^n$ at locations $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)^\top \in \mathbb{R}^{n\times p}$. We place a GP prior on the function f that models all function values as jointly Gaussian, with a covariance $\mathcal{K} : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$. We use the notation $\mathbf{f} = f(\mathbf{X})$ and $\mathcal{K}(\mathbf{X}, \mathbf{X})_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$. Then, the joint density for $\mathbf{y}$ and $\mathbf{f}$ for a single-layer Gaussian process (GP) is $p(\mathbf{y}, \mathbf{f}) = p(\mathbf{f}; \mathbf{X}) \prod_{i=1}^n p(y_i|f_i)$, where $\mathbf{f}|\mathbf{X} \sim N(0, \mathcal{K}(\mathbf{X}, \mathbf{X}))$ and $y_i|f_i \sim N(f_i, \delta^2)$.

For L vector-valued stochastic functions denoted as $\mathcal{F}^\ell$, a deep Gaussian process (DGP) Damianou and Lawrence (2013) defines a prior recursively on $\mathcal{F}^1, ..., \mathcal{F}^L$. The prior on each function $\mathcal{F}^\ell$ is an independent GP in each dimension, with input locations given by the function values at the previous layer: the outputs of GPs at layer ℓ are $\{\mathbf{F}^\ell_{.j}\}_{j=1}^{d}$ and the corresponding inputs are $\mathbf{F}^{\ell-1}$. The joint density is

$$p(\mathbf{y}, \{\mathbf{F}^\ell\}_{\ell=1}^L) = \prod_{i=1}^{n} p(y_i|f_i^L) \prod_{\ell=1}^{L} p(\mathbf{F}^\ell|\mathbf{F}^{\ell-1}),$$

where $\mathbf{F}^0 = \mathbf{X}$, $\mathbf{F}^\ell \in \mathbb{R}^{n\times d_\ell}$ for $0 < \ell \leqslant L$, $\mathbf{F}^\ell_{.j}|\mathbf{F}^{\ell-1} \sim N(0, \mathcal{K}^\ell_j(\mathbf{F}^{\ell-1}, \mathbf{F}^{\ell-1}))$ for $1 \leqslant j \leqslant d_\ell$, $0 < \ell \leqslant L$.

Variational inference (VI) for Bayesian models. Consider the joint density of the latent variables $\mathbf{f} = \{f_i\}_{i=1}^m$ and the observations $\mathbf{y} = \{y_i\}_{i=1}^n$, $p(\mathbf{f}, \mathbf{y}) = p(\mathbf{f})p(\mathbf{y}|\mathbf{f})$. We know that inference in any Bayesian model amounts to conditioning on the data and computing the posterior $p(\mathbf{f}|\mathbf{y})$. In models like DGP, this calculation is difficult and so, we use approximate inference. A popular strategy is variational inference (VI) Blei et al. (2017) which requires specifying a family of *approximate* densities $\mathcal{Q}$. Our goal is to find the member q^* of that family which minimizes the Kullback-Leibler (KL) divergence to the exact posterior,

$$q^*(\mathbf{f}) = \arg\min_{q(\mathbf{f})\in\mathcal{Q}} \mathbf{KL}(q(\mathbf{f})||p(\mathbf{f}|\mathbf{y})).$$

Instead of minimizing the KL divergence, one maximizes the evidence lower bound (ELBO),

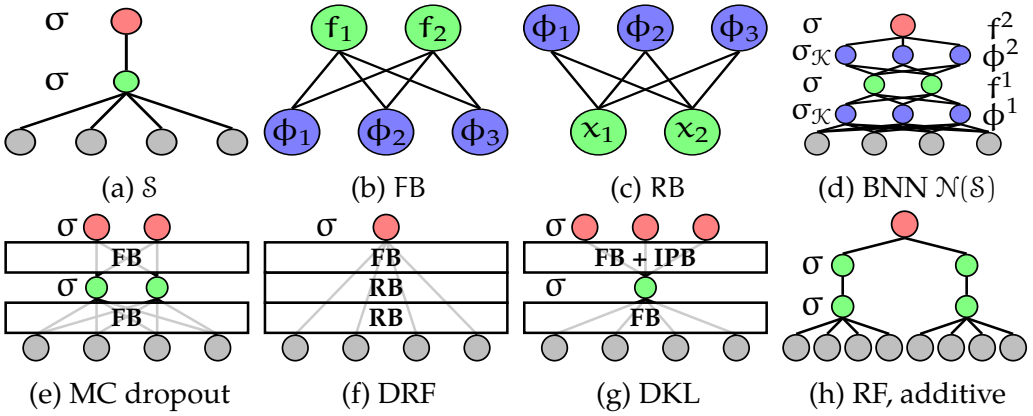
$$\mathbf{ELBO}(q) = \mathbb{E}_{q(\mathbf{f})}[\log p(\mathbf{y}|\mathbf{f})] - \mathbf{KL}(q(\mathbf{f}||p(\mathbf{f})).$$

The first term is an expected likelihood, which encourages the densities to place their mass on configurations of the latent variables that explain the observed data. The second term is the negative KL divergence between the variational density and the prior so the densities lie close to the prior.

Analysis of variance (ANOVA) decomposition. Consider the inference task for a stochastic function f with a set of n observations $(y_1, ..., y_n)^\top$ at locations $(\mathbf{x}_1, ..., \mathbf{x}_n)^\top$. In a simple setting, we assume that $p(y|f)$ follows the Gaussian distribution with $y_i = f(\mathbf{x}_i) + \epsilon_i,\ \epsilon_i \sim N(0, \delta^2)$. Based on Gu and Wahba (1993), for a multivariate function f, $f(\mathbf{x}) = \left[\prod_{j=1}^p (I - E_j + E_j)\right] f(\mathbf{x})$. Expanding the product, we obtain the equivalent representation for f in an ANOVA decomposition,

$$f(\mathbf{x}) = f_0 + \sum_{j=1}^p f_j(x^j) + \sum_{j=1}^p \sum_{k=1}^p f_{j,k}(x^j, x^k) + \cdots, \tag{5.1}$$

where $f_0 = (\prod_{j=1}^p E_j) f(\mathbf{x})$ is the mean constant, $f_1(x^1) = (I - E_1)(\prod_{j\neq 1} E_j) f(\mathbf{x})$ is the main effect function for the first dimension x^1 of $\mathbf{x}$ and $f_{1,2}(x^1, x^2) = (I - E_1)(I - E_2)(\prod_{j\neq 1,2} E_j) f(\mathbf{x})$ is the second-order interaction function between the first



(a) $\mathcal{S}$ (b) FB (c) RB (d) BNN $\mathcal{N}(\mathcal{S})$

(e) MC dropout (f) DRF (g) DKL (h) RF, additive

Figure 5.1: Using $\mathcal{S}$ in (a), one can construct a BNN $\mathcal{N}(\mathcal{S})$ in (d) with the function block (FB) in (b) and the random feature block (RB) in (c). For different $\mathcal{S}$, (e) is MC dropout, (f) is deep random features, (g) is deep kernel learning. (g) also represents a multi-task $\mathcal{S}$ while (h) gives the additive structure.

dimension x^1 and the second dimension x^2 of $\mathbf{x}$. The rest of the terms can be defined similarly. The ANOVA decomposition makes the prediction interpretable by splitting f into sub-functions, such as f_1 and $f_{1,2}$: we can understand the main effect of a single parameter or interactions between a few parameters.

5.2 Deep Probabilistic Models with a Computation Skeleton Assumption

We first define the computation skeleton and show how it imposes structure on the function class. This will directly facilitate understanding flow of information from input to output. Later, we show that the constructed BNNs can be seen as VI approximations for the DGPs with the expressive power of the kernel. Finally, we discuss how to understand other deep probabilistic models Gal and Ghahramani (2016); Wilson et al. (2016); Daniely et al. (2016) in our framework.

What is a computation skeleton? Computation skeleton Daniely et al. (2016) is a gadget to compactly describe a feed-forward computation structure from the inputs to the outputs: in other words, the *flow of information*. Formally, a computation

skeleton $\mathcal{S}$ is a multi-layer graph with the bottom nodes representing inputs, the top nodes representing outputs and non-input nodes are labeled by activations σ. In Daniely et al. (2016), this idea was used to study a family of DNNs: it was shown that DNNs can be seen as the realization of certain types of structures and their dual kernels. In fact, every $\mathcal{S}$ defines a specific NN: Fig. 5.1(a) shows a two layer fully connected NN, Daniely et al. (2016) shows more examples. Here, we reuse the name but define a slightly different $\mathcal{S}$ to restrict the function class. For notational simplicity, we consider $\mathcal{S}$'s with a single output.

What are blocks? To express a deep probabilistic model (say, BNNs) using a computation skeleton $\mathcal{S}$, we also need two additional components, which we call "blocks". Our first type of block is a **function block** denoted as $\mathtt{FB}(\mathbb{P}_{\mathbf{v}}, r, d)$, which allows every "node" in $\mathcal{S}$ to replicate d times. This will help us in defining Bayesian priors and posteriors. We setup FB as a one layer NN where the inputs nodes and output nodes are fully connected. All incoming edges to the output node f_j as in Fig. 5.1(b) form a vector $\mathbf{v}_j$. The set of $\mathbf{v}_j$'s for $1 \leqslant j \leqslant d$ i.i.d. follow the distribution $\mathbb{P}_{\mathbf{v}}$ on $\mathbb{R}^r$. $\mathtt{FB}(\mathbb{P}_{\mathbf{v}}, r, d)$ simply takes the inputs $\boldsymbol{\phi} = (\phi_1, ..., \phi_r)^T$ and outputs a d-dimension vector $\mathbf{f}$ with $f_j = \boldsymbol{\phi}^T \mathbf{v}_j$ for $1 \leqslant j \leqslant d$. Our second type of block is a **random feature block** denoted as $\mathtt{RB}(\mathbb{P}_{\mathbf{w}}, d, r, \sigma_{\mathcal{K}})$, which we use to construct random feature approximations for kernels to leverage the expressive power of DGP. We setup RB as a one layer NN with random weights where the inputs nodes and outputs are fully connected. All incoming edges to the output node ϕ_j as in Fig. 5.1(c) form a vector $\mathbf{w}_j$. The set of $\mathbf{w}_j$'s for $1 \leqslant j \leqslant r$ follow the distribution $\mathbb{P}_{\mathbf{w}}$ on $\mathbb{R}^d$ for $1 \leqslant j \leqslant d$. $\mathtt{RB}(\mathbb{P}_{\mathbf{w}}, d, r, \sigma_{\mathcal{K}})$ takes the inputs $\mathbf{x} = (x_1, ..., x_d)^T$ and outputs a r-dimension vector $\boldsymbol{\phi}$ with $\phi_j = \frac{1}{\sqrt{r}} \sigma_{\mathcal{K}}(\mathbf{x}^T \mathbf{w}_j)$ for $1 \leqslant j \leqslant r$.

Writing a BNN with $\mathcal{S}$, FB and RB blocks. Let us denote s^ℓ to be the number of nodes in layer ℓ of the computation skeleton $\mathcal{S}$. Typically, we may choose $\mathbb{P}_{\mathbf{v}} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\mathbb{P}_{\mathbf{w}} \sim \rho N(\mathbf{0}, \mathbf{I})$ for a constant ρ. Alg. 3 shows how given a $\mathcal{S}$, together with FB and RB blocks, we can construct a BNN $\mathcal{N}(\mathcal{S})$ by sequentially replacing edges in $\mathcal{S}$ with a combination of FB and RB from bottom (input nodes) up to the top (output nodes). Shortly, we describe the properties of such a BNN. First, let us see an example. For Fig. 5.1, using FB in (b) with $r = 3$ and $d = 2$ ($d = 1$ for the

last layer) and RB in (c) with $d = 2$ and $r = 3$ ($d = 4$ for the first layer) in Algorithm 3, we construct a BNN in (d) from the in $\mathcal{S}$ in (a). Essentially, we "substitute in" FB + RB to replace every edge in Fig. 5.1(a).

Prior and Posterior Approximation for $\mathcal{N}(\mathcal{S})$

Our remaining task is to describe a prior for $\mathcal{N}(\mathcal{S})$ and then derive a posterior approximation scheme for the construction in Alg. 3. To do so, we define some notations. We use $\mathbf{W}$ for all random weights in the RB blocks, $\mathbf{V}$ gives all BNN weights in the FB blocks and $\mathbf{v}_k^\ell$ denotes the weight vector that goes into kth dimension of $\mathbf{f}^\ell$. The related random features are denoted by $\boldsymbol{\phi}_k^\ell$. For $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)^T$, we denote $\mathbf{F}^\ell$ as a matrix with the ith row $\mathbf{F}_{i\cdot}^\ell = \mathbf{f}^\ell(\mathbf{x}_i)$ as the value of $\mathbf{f}^\ell$ evaluated on input $\mathbf{x}_i$. We define $\boldsymbol{\Phi}_k^\ell$ to be the random feature matrix related to $\mathbf{f}_k^\ell$ for $1 \leqslant k \leqslant d^\ell$.

Definition 5.1. *For a BNN $\mathcal{N}(\mathcal{S})$ from Algorithm 3, we treat $\mathbf{W}$ as fixed, then the parameters are only $\mathbf{V}$. We choose $\mathbb{P}_\mathbf{v} = N(\mathbf{0}, \mathbf{I}_r)$ in Algorithm 3 to define the Bayesian prior on $\mathbf{v}_k^\ell$ as $N(\mathbf{0}, \mathbf{I}_r)$ for $1 \leqslant k \leqslant d^\ell$, $1 \leqslant \ell \leqslant L$. This Bayesian prior leads to the relation $p(\mathbf{F}_{\cdot k}^\ell | \mathbf{F}^{\ell-1}) = N(\mathbf{F}_{\cdot k}^\ell; \mathbf{0}, \boldsymbol{\Phi}_k^\ell \boldsymbol{\Phi}_k^{\ell\,T})$, therefore has a distribution over $\{\mathbf{F}^\ell\}_{\ell=1}^L$ which is $p(\{\mathbf{F}^\ell\}_{\ell=1}^L) = \prod_{\ell=1}^L p(\mathbf{F}^\ell | \mathbf{F}^{\ell-1})$.*

Algorithm 3 Expressing a Bayesian neural network (BNN) with computation skeleton and blocks

Input: a computation skeleton $\mathcal{S}$. **Output:** a deep BNN $\mathcal{N}(\mathcal{S})$.
Construct layer 0 in $\mathcal{N}(\mathcal{S})$ by copying inputs (layer 0) from $\mathcal{S}$.
for $\ell = 1$ to L **do**
  $\mathbf{f}^{\ell-1} = (\mathbf{f}_1^{\ell-1}; ...; \mathbf{f}_{s^{\ell-1}}^{\ell-1}) \in \mathbb{R}^{d^{\ell-1}}$: output vector on layer $\ell - 1$ in $\mathcal{N}(\mathcal{S})$.
  For each $\mathbf{f}_j^{\ell-1}$, $1 \leqslant j \leqslant s^{\ell-1}$, apply the activation σ in $\mathcal{S}$, and output $\{\sigma(\mathbf{f}_j^{\ell-1})\}_{j=1}^{s^{\ell-1}}$.
  for $i = 1$ to s^ℓ **do**
    $In(i) = \{1 \leqslant j \leqslant s^{\ell-1} |$ if node j in layer $\ell - 1$ connects with node i in layer ℓ in $\mathcal{S}\}$
    Build $RB(\mathbb{P}_{\mathbf{w}_i^\ell}, d^{\ell-1}, r, \sigma_\mathcal{K})$ on $\{\sigma(\mathbf{f}_j^{\ell-1})\}_{j \in In(i)}$ and output $\boldsymbol{\phi}_i^\ell \in \mathbb{R}^r$.
    Build $FB(\mathbb{P}_{\mathbf{v}_i^\ell}, r, d_i^\ell)$ on $\boldsymbol{\phi}_i^\ell$ and output $\mathbf{f}_i^\ell \in \mathbb{R}^{d_i^\ell}$ in layer ℓ of $\mathcal{N}(\mathcal{S})$
  end for
end for

When the outputs $\mathbf{y}$ and likelihood $p(\mathbf{y}|\mathbf{F}^L)$ are available for the design matrix $\mathbf{X}$, the posterior of BNN $\mathcal{N}(\mathcal{S})$ is intractable. Therefore, we use variational inference to approximate its posterior.

Definition 5.2. *We define the variational inference approximation for the posterior of* $\mathbf{V}$ *in* $\mathcal{N}(\mathcal{S})$ *by defining the variational posterior* q *over* $\mathbf{V}$ *with* $\mathbf{v}_k^\ell \sim N(\boldsymbol{\mu}_k^\ell, \boldsymbol{\Sigma}_k^\ell)$*. Then, we get the* ***ELBO*** $= \sum_{i=1}^n \mathbb{E}_{q(f_i^L)} \log(p(y_i|f_i)) - \boldsymbol{KL}(q(\mathbf{V})|p(\mathbf{V}))$*. This variational posterior over* $\mathbf{V}$ *also leads to a posterior over* $\{\mathbf{F}^\ell\}_{\ell=1}^L$*.*

To optimize the ELBO, we apply a doubly stochastic approximation for the first term in the ELBO, where the sum is estimated using mini-batches and the expectation is approximated with a Monte Carlo sample from the variational posterior $q(f_i^L)$. Both stochastic approximations are unbiased. Further, by reparameterizing $\mathbf{v}_k^\ell = \boldsymbol{\mu}_k^\ell + {\boldsymbol{\Sigma}_k^\ell}^{1/2} N(\mathbf{0}, \mathbf{I}_r)$, the optimization of ELBO can be achieved with mini-batch training and backpropagationCutajar et al. (2017); Salimbeni and Deisenroth (2017); Gal and Ghahramani (2016).

Relationship between $\mathcal{N}(\mathcal{S})$ and DGPs

Having constructed a BNN $\mathcal{N}(\mathcal{S})$ from $\mathcal{S}$, now we show that it can be seen as a VI approximation for the DGP. To simplify notation, we assume that all $\{\boldsymbol{\Phi}_k^\ell\}_{k=1}^{d^\ell}$ are the same so we drop the subscript k. We also assume that all $\{d^\ell\}_{\ell=1}^L$ are the same. We define an empirical kernel $\hat{\mathcal{K}}^\ell(\mathbf{f}^{\ell-1}(\mathbf{x}), \mathbf{f}^{\ell-1}(\mathbf{x}'))$ as $\frac{1}{r}\sum_{i=1}^r \sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))^T\mathbf{w}_i)\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'))^T\mathbf{w}_i)$ and it's expectation $\mathcal{K}^\ell(\mathbf{f}^{\ell-1}(\mathbf{x}), \mathbf{f}^{\ell-1}(\mathbf{x}'))$ as $\mathbb{E}_{\mathbf{w}}\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(x))^T\mathbf{w})\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(x'))^T\mathbf{w}$.

It is easy to check that $\hat{\mathcal{K}}^\ell(\mathbf{f}^{\ell-1}(\mathbf{x}), \mathbf{f}^{\ell-1}(\mathbf{x}')) = \langle \boldsymbol{\phi}^\ell(\mathbf{x}), \boldsymbol{\phi}^\ell(\mathbf{x}')\rangle$. We denote $\hat{\mathcal{K}}^\ell(\mathbf{F}^{\ell-1}, \mathbf{F}^{\ell-1})$ as the $n \times n$ matrix for n inputs. First, we point out that the prior in Definition 5.1 is indeed a DGP prior.

Proposition 5.3. *The BNN prior of* $\mathcal{N}(\mathcal{S})$ *in Def. 5.1 represents a DGP prior for* $\{\mathbf{F}^\ell\}_{\ell=1}^L$*. This means that* $\mathbf{F}_{\cdot j}^\ell|\mathbf{W}, \mathbf{F}^{\ell-1} \sim N(0, \hat{\mathcal{K}}^\ell(\mathbf{F}^{\ell-1}, \mathbf{F}^{\ell-1}))$*, for* $1 \leqslant j \leqslant d$ *and* $1 \leqslant \ell \leqslant L$*.*

This DGP with kernels $\{\hat{\mathcal{K}}^\ell\}_{\ell=1}^L$ is also an approximation for the DGP with kernels $\{\mathcal{K}^\ell\}_{\ell=1}^L$ if $\sigma_{\mathcal{K}}$ is ReLU or C-bounded ($\sigma_{\mathcal{K}}$ is continuously differentiable and $\|\sigma_{\mathcal{K}}\|_\infty, \|\sigma'_{\mathcal{K}}\|_\infty \leqslant C$).

C-boundedness. The C-bounded condition holds for most popular sigmoid-like functions such as $1/(1+e^{-x}), \mathrm{erf}(x), x/\sqrt{1+x^2}, \tanh(x)$ and $\tan^{-1}(x)$. The bound for C-bounded activation functions is given in the supplement, which is similar to the ReLU results below.

Theorem 5.4. *If the activation function $\sigma_{\mathcal{K}}$ is ReLU, then for every $1 \leqslant \ell \leqslant L$, on a compact set $\mathcal{M} \in \mathbb{R}^d$ with diameter $\mathtt{diam}(\mathcal{M})$ and $\max_{\Delta\in\mathcal{M}} \|\Delta\|_2 \leqslant c_{\mathcal{M}}$, with probability at least $1 - c_1 c_{\mathcal{M}} \mathtt{diam}(\mathcal{M})^2 exp\left\{-\frac{r\epsilon^2}{8(1+d)\nu_{\mathcal{M}}^2}\right\}$, for any $\sigma(\mathbf{f}^{\ell-1}(\mathbf{x})), \sigma(\mathbf{f}^{\ell-1}(\mathbf{x}')) \in \tilde{\mathcal{M}}$,*

$$|\hat{\mathcal{K}}^{\ell}(\mathbf{f}^{\ell-1}(\mathbf{x}), \mathbf{f}^{\ell-1}(\mathbf{x}')) - \mathcal{K}^{\ell}(\mathbf{f}^{\ell-1}(\mathbf{x}), \mathbf{f}^{\ell-1}(\mathbf{x}'))| \leqslant \epsilon,$$

for a constant $c_1 > 0$ and a parameter $\nu_{\mathcal{M}}$ depending on $\mathcal{M}$. Here, $\tilde{\mathcal{M}}$ specifies that we require $\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))$ and $\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'))$ to be two vectors in $\mathcal{M}$ that are not collinear.

Remark 5.5. *In Cho and Saul (2009); Cutajar et al. (2017), the authors point out that $\mathcal{K}^{\ell}$ is the arc-cosine kernel for ReLU activation function and in Cutajar et al. (2017), the authors use this random feature approximation idea to construct a BNN. In this direction of works, our main contribution is the theoretical analysis. In Daniely et al. (2016), the author proves the convergence rate for $\hat{\mathcal{K}}$ given fixed points $\mathbf{x}$ and $\mathbf{x}'$. Theorem 5.4 is stronger which proves the uniform convergence rate on a compact set $\mathcal{M}$. It implies the effect of the diameter $\mathtt{diam}(\mathcal{M})$, the angle between $\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))$ and $\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'))$, and the effect of the replication d of FB block besides the effect of the RB block replication r.*

We show that the BNN prior of $\mathcal{N}(\mathcal{S})$ in Def. 5.1 is a prior for DGP with kernels $\{\hat{\mathcal{K}}^{\ell}\}_{\ell=1}^{L}$, which is also an approximation for DGP with kernels $\{\mathcal{K}^{\ell}\}_{\ell=1}^{L}$. Next, we show that the variational posterior and ELBO optimization in Definition 5.2 leads to the same posterior approximation for DGP as using a recent "inducing points" approach via doubly stochastic variational inference Salimbeni and Deisenroth (2017).

Theorem 5.6. *Using inducing points with doubly stochastic variational inference Salimbeni and Deisenroth (2017) to approximate the posterior of the DGP with kernels $\{\hat{\mathcal{K}}^{\ell}\}_{\ell=1}^{L}$, we*

obtain exactly the same posterior approximation as solving the ELBO optimization in Definition 5.2 for $\mathcal{N}(\mathcal{S})$.

Remark 5.7 (Proof discussion)**.** *First, since we show that the prior of* $\mathcal{N}(\mathcal{S})$ *in Def. 5.1 is a prior for DGP with kernels* $\{\hat{\mathcal{K}}^\ell\}_{\ell=1}^{\mathrm{L}}$*, we can apply the inducing points approximation for DGP. Second, the variational posterior defined in Definition 5.2 is over* $\mathbf{v}^\ell$ *while the variational posterior from inducing points approach is defined over* $\mathbf{F}^\ell$ *for each layer* ℓ*. We need to show the equivalence for variational posterior and ELBO between the two approaches. Then, because the feature space for the kernel* $\{\hat{\mathcal{K}}^\ell\}_{\ell=1}^{\mathrm{L}}$ *is finite with rank* r*, by randomly selecting* r *inducing points, we can decompose the kernel matrix into a product of two square matrix, which is also invertible based on random matrix theory Rudelson (2008); Tao (2012). Finally, we simplify both sides and we achieve the equivalence. More details are in the supplement.*

Based on Prop. 5.3 and Thm. 5.6, we show that the BNN $\mathcal{N}(\mathcal{S})$ constructed from $\mathcal{S}$ can be seen as a VI approximation for the DGP with kernels $\{\hat{\mathcal{K}}^\ell\}_{\ell=1}^{\mathrm{L}}$. However, we notice that the empirical kernel $\{\hat{\mathcal{K}}^\ell\}_{\ell=1}^{\mathrm{L}}$ and its expectation kernel are restricted by the class of $\sigma_{\mathcal{K}}$ which does not cover all general kernels. We solve this problem by proposing a new block in BNNs, which is named as **inducing points block** IPB.

Definition 5.8. *For a kernel* $\mathcal{K}$*,* IPB *can be constructed by choosing* r *additional points* $\mathbf{Z}$ *(inducing points), taking the inputs* $\mathbf{x}$ *and outputting an* r*-dimension vector* $\mathcal{K}(\mathbf{x}, \mathbf{Z})\mathcal{K}(\mathbf{Z}, \mathbf{Z})^{-1/2}$.

Replacing RB in Alg. 3 by IPB, we obtain a new class of BNNs. We prove that the $\mathcal{N}(\mathcal{S})$ with IPB can be seen as a VI approximation for DGP with general kernels $\{\mathcal{K}^\ell\}_{\ell=1}^{\mathrm{L}}$.

Theorem 5.9. *Using inducing points with doubly stochastic variational inference Salimbeni and Deisenroth (2017) to approximate the posterior of the DGP with kernels* $\{\mathcal{K}^\ell\}_{\ell=1}^{\mathrm{L}}$*, we obtain the same posterior approximation as solving the ELBO optimization in Definition 5.2 for* $\mathcal{N}(\mathcal{S})$ *with* IPB*, except a constant offset in each layer (see supplement).*

Theorem 5.9 proves that for any DGP with a general kernel $\{\mathcal{K}^\ell\}_{\ell=1}^{\mathrm{L}}$, the $\mathcal{N}(\mathcal{S})$ with IPB can be seen as a VI approximation. The similarity between RB matrix $\mathbf{\Phi}$ and

the IPB matrix $\mathcal{K}(\mathbf{x}, \mathbf{Z})\mathcal{K}(\mathbf{Z}, \mathbf{Z})^{-1/2}$ is that they are both rank r basis approximation for the kernel $\mathcal{K}$, which are equivalent when the kernel is $\hat{\mathcal{K}}$ related to activation function $\sigma_{\mathcal{K}}$.

$\mathcal{N}(\mathcal{S})$ and Other Deep Probabilistic Models

We have shown that the BNN $\mathcal{N}(\mathcal{S})$ constructed from Alg. 3 can be seen as a VI approximation for DGP. Next, we show that with a few small changes, Alg. 3 can construct other interesting deep probabilistic models. To do so, we allow changes in the existing framework,

Change 1) In Alg. 3, inside the inner-most loop, we have one RB (IPB). We allow taking out RB (IPB) entirely or replacing it by multiple sequential RBs (IPBs) as long as they are matched.

Change 2) Earlier, we assumed that the variational posterior q for $\mathbf{v}_i^\ell$ follows a normal distribution. We now allow it to follow a probability mass function and a mixture of two probability mass functions.

Change 3) Earlier in Def. 5.1, the prior $p(\mathbf{v})$ is a normal distribution. We allow other forms of priors to encourage other regularizations, e.g., Laplace distribution for ℓ_1 sparsity.

Remark 5.10. *Change 1 determines the complexity of kernels in each layer. Since the exact posterior can be multi-modal, Change 2 allows us to use different variational posterior class from normal distribution. As prior in Bayes often serves as a regularization, Change 3 allows us to choose priors related to Lasso type penalty while normal distribution corresponds to* L_2 *penalty.*

By simply applying these changes, $\mathcal{N}(\mathcal{S})$ can lead to most popular deep probabilistic models. **First**, consider the case where no RB is used in constructing $\mathcal{N}(\mathcal{S})$. This gives us a kernel $\mathcal{K}$ with $\mathcal{K}(\mathbf{x}, \mathbf{x}') = \sigma(\mathbf{x})^\mathsf{T}\sigma(\mathbf{x}')$ as in Fig. 5.1 (e). Further, let us use $q(\mathbf{v})$ as a mixture of two probability mass functions. These changes leads to MC dropout Gal and Ghahramani (2016). **Second**, let us allow multiple RBs in constructing $\mathcal{N}(\mathcal{S})$ as in Fig. 5.1(f). This ends up representing the Gaussian

process with deep random feature in Daniely et al. (2016). **Third**, in Fig.5.1 (g), we show another construction that represents deep kernel learning Wilson et al. (2016), where IPB or RB is not used at all; the variational posterior $q(\mathbf{v})$ is a probability mass function except the last layer.

Remark 5.11. *One of the contribution from our framework is that it ties most deep probabilistic models to study their similarity and difference. It shows the similarity among them through the computation skeleton $\mathcal{N}(\mathcal{S})$ in Alg. 3. It distinguishes them through blocks and simple changes. Through our framework, it is easy to compare different deep probabilistic models Salimbeni and Deisenroth (2017); Cutajar et al. (2017); Gal and Ghahramani (2016); Daniely et al. (2016); Wilson et al. (2016) and derive new constructions, approximation methods and optimization algorithms.*

Computation skeleton and structure. We see that the structure assumption is applied via the computation skeleton and we can construct various deep probabilistic models by adjusting blocks in Alg. 3. A computation skeleton, which has a meaningful structure for the applications, can help to build a deep probabilistic model that is interpretable. For example, in Fig. 5.1 (g), we see a computation skeleton for multi-task learning where the first layer captures the shared low level patterns among multiple tasks and the second layer defines individual high level patterns for each task. In Fig. 5.1 (h), we have an additive structure where a large neural network is composed by summing several sub neural networks, which helps to simplify the prediction model and understand the interactions within small input groups.

5.3 Additive Structure and Interactions

Let us see an example of constructing BNNs with additive structure in our framework to detect statistical interactions and provide interpretability. In eq.(5.1), we introduced the ANOVA decomposition, which represents a complicated function $f(\mathbf{x})$ as an additive model of several sub-functions. Each sub-function represents an interaction term. For example, given a function f^* between inputs $\mathbf{x} = (x_1, ..., x_p)$

and output y, one can define the interaction function $\mathbf{I}_T$ over a subset of inputs $\mathbf{x}_T$ Gu and Wahba (1993) as

$$\mathbf{I}_T(\mathbf{x}_T) = \prod_{i \in T} (I_{x_i} - \mathbb{E}_{x_i}) \prod_{j \notin T} \mathbb{E}_{x_j} f^*(x_1, ..., x_p). \tag{5.2}$$

The ANOVA decomposition helps us understand the flow of information from the input to the output. For example, $\mathbf{I}_1(\mathbf{x}_{\{1\}})$ is the main effect of first input $\mathbf{x}_1$ and $\mathbf{I}_{\{1,2\}}(\mathbf{x}_{\{1,2\}})$ is the interaction between the first two inputs.

First, we discuss how to design a DNN with the additive structure and detect statistical interactions. Then using Alg. 3, the results can be easily extended to the Bayesian formulation which yields a Bayesian additive neural network. Based on Fig. 5.1(h), the additive neural network $f(\mathbf{x})$ is $f(\mathbf{x}) = \sum_{j=1}^{k} g_j(\mathbf{x})$, where each sub-function $g_j(\mathbf{x})$ is a sub-neural network with few inputs. In the smoothing spline ANOVA model Gu and Wahba (1993), one considers all possible sub-functions in ANOVA, which means that the number of components k is 2^p. The sub-function $g_j(\mathbf{x})$ is a non-parametric model in a reproducing kernel Hilbert space defined by a kernel $\mathcal{K}_j(\mathbf{x})$. However, when high-order interactions exist, the kernel is not complex enough to model the interactions and considering all possible sub-functions in the ANOVA decomposition is expensive. Since each sub-function $g_j(\mathbf{x})$ in our additive neural network is a neural network, it has the capability to model complex interactions even when the dimension is high. We choose k to be a polynomial number of p. In order to generate similar results as the ANOVA decomposition to study interactions, we use a post-training estimate. After we fit a function $f(\mathbf{x})$ from the training step, we apply the empirical expectation operator $\mathbb{E}^n$ with n samples on $f(\mathbf{x})$,

$$\mathbf{I}_T^n(\mathbf{x}_T) = \prod_{i \in T} (I_{x_i} - \mathbb{E}_{x_i}^n) \prod_{j \notin T} \mathbb{E}_{x_j}^n f(x_1, ..., x_p), \tag{5.3}$$

which is an approximation for the interaction function $\mathbf{I}_T(\mathbf{x}_T)$. Detecting statistical interactions is hard when p is large, but we show that our additive neural network model with the post-training ANOVA decomposition in (5.3) can estimate interactions in polynomial complexity of p.

Theorem 5.12. *If there exist inputs clusters* $\{T_j^*\}_{j=1}^{k^*}$ *such that* $f^*(\mathbf{x}) = \sum_{j=1}^{k^*} g_j^*(\mathbf{x}_{T_j^*})$ *with* k^* *of the order of a polynomial in* p *and* $c = \max_{j=1}^{k^*} |T_j^*| = O(\log p)$*, then there exists a trained additive neural network that predicts* $\mathbf{y}$ *well and restricts the number of possible interactions to be at most a polynomial in* p*. Further, if every sub neural network has* L *layers with* d *hidden units, then the complexity of* (5.3) *is at most* $n^c k^* d^{2L-1}$*, which is also polynomial in* p*.*

Remark 5.13. *This result does not hold for an arbitrary neural network, which needs to consider all* 2^p *possible interactions in the post-training ANOVA decomposition and needs* $n^p (k^* d)^{2L-1}$ *complexity for* (5.3)*. When computing* (5.3)*, an additive neural network is more efficient since each block is more compact, and the additive structure helps eliminate non-existing interactions from the candidates set. Therefore, when the true function* f^* *has a good additive representation, additive neural network is more efficient. In practice, we always use group Lasso type penalty to encourage each sub neural network to depend on a few inputs.*

Bayesian additive neural network (BANN). We present the computation skeleton of the additive structure in Fig. 5.1 (h). By applying our BNN construction framework in Alg. 3, we can easily construct the Bayesian formulation of our additive neural network model using various deep probabilistic models. In order to allow each sub-neural network selects a subset of inputs, for every sub-neural network, the first layer is only built with FB and the prior on the weights is $p(\mathbf{V}^1) \sim \exp(-\sum_{i=1}^{p} \|\mathbf{v}_i^1\|_2)$ where $\mathbf{v}_i^1$ refers to the weight vector emanating from the ith input. The variational posterior $q(\mathbf{V}^1)$ is the probability mass function to make top layers stable. Other layers are constructed depending on the specific deep probabilistic model in Alg. 3. That generates a Bayesian additive neural network which can estimate interaction, provide interpretability and derive uncertainties for measures of interest.

5.4 Experiments

We show how our BANN model can detect statistical interactions and how one can easily compare various deep probabilistic models. We first evaluate the performance of our BANN model on synthetic experiments for regression and interaction detection for additive functions. Then, we use Alg. 3 to construct four different types of BANNs (where each refers to a type of deep probabilistic model) and check its utility for prediction and identifying interaction strength. Finally, we show how BANN can infer main effects and statistical interactions of features with uncertainties for interpretability. Further, we apply BANN on eight benchmark datasets to show its performance.

Table 5.1: (L) Synthetic functions from Tsang et al. (2018); (R) Comparisons between BANN, BNN, BART and NID. BNN/BART do not detect interactions.

Method	formula	RMSE				Top rank recall (noise, $\sigma^2 = 1, 3, 5$)					
		BANN (0.5k)	BNN (7k)	BART	NID (20k)	**Ours** (BANN)			NID		
f_1	$10\sin(\pi x_1 x_2) + 20(x_3 - .5)^2$	1.07 ± 0.01	1.15 ± 0.01	1.07 ± 0.01	1.09 ± 0.01	**1**	**1**	**1**	1	1	1
f_2	$10\exp(x_1 x_2) - 20\cos(x_3 + x_4 + x_5)$ $+7\arcsin(x_9 x_{10})$	1.16 ± 0.01	1.22 ± 0.02	1.43 ± 0.02	1.44 ± 0.02	**1**	**1**	**2/3**	1	2/3	0
f_3	$\exp(\lvert x_1 x_2 \rvert + 1) + \exp(\lvert x_3 + x_4 \rvert + 1)$ $-19\cos(x_5 + x_6) - 10\sqrt{x_8^2 + x_9^2 + x_{10}^2}$	1.35 ± 0.01	1.32 ± 0.01	1.24 ± 0.02	1.42 ± 0.02	3/4	**3/4**	**2/4**	1	2/4	1/4
f_4	$\frac{1}{1+x_1^2+x_2^2+x_3^2} - 5\sqrt{\exp(x_4 + x_5)}$ $+10\lvert x_6 + x_7 \rvert + 6x_8 x_9 x_{10}$	1.13 ± 0.01	1.13 ± 0.01	1.17 ± 0.01	1.40 ± 0.02	**3/4**	**3/4**	**2/4**	2/4	2/4	1/4

BANN, BNN, BART and NID on prediction accuracy and interaction detection. We compare BANN with BNN (with a single neural network), BART (Bayesian additive regression tree) and NID (Neural interaction detection) in terms of prediction accuracy and interaction detection. For BANN, we use the setup in section 5.3, where the group Lasso penalty is applied on the first layer. We use 10 compact sub-NNs for BANN and a single (but more complex) neural network for BNN (see supplement). For BART and NID, we use the setup in Chipman et al. (2010); Tsang et al. (2018). Both BANN and BNN here are based on the MC dropout type construction (see section 5.2). First, we compare RMSE (root mean-squared-error). We run 4 synthetic experiments, see Tab. 5.1. We use one function f in Tab. 5.1 to generate 5000 train/test samples (10 features, 1 response), where for every input $\mathbf{x}$, each dimension of the inputs are i.i.d. generated from the uniform distribution on $(0, 1]$ and the response y is $y = f(\mathbf{x}) + \epsilon$, with $\epsilon \sim N(0, 1)$. From Tab. 5.1, we see

that the BANN yields comparable (or better) RMSE compared to baselines. Though the prediction performance is similar, note that BANN is a much more compact design: BANN has just ~500 edges while the BNN has 7000 edges, NID has 2000 edges and BART has 200 trees (see Tab. 5.1).

Table 5.2: Constructing multiple types BANN of f_1.

Measure	MLL	Interaction	Main effect				
		$(1,2)$	1	2	3	4	5
MC dropout	-1.61 ± 0.09	1.51 ± 0.05	2.44 ± 0.15	2.35 ± 0.10	1.69 ± 0.06	3.18 ± 0.04	1.63 ± 0.03
RF	-1.60 ± 0.09	1.52 ± 0.06	2.39 ± 0.08	2.31 ± 0.11	1.70 ± 0.11	3.19 ± 0.06	1.61 ± 0.04
DKL	-1.53 ± 0.08	1.59 ± 0.04	2.44 ± 0.23	2.32 ± 0.13	1.70 ± 0.08	3.16 ± 0.06	1.61 ± 0.04
DRF	-1.56 ± 0.07	1.40 ± 0.02	2.36 ± 0.05	2.35 ± 0.10	1.71 ± 0.03	3.15 ± 0.03	1.59 ± 0.02

Next, we compare BANN and NID for interaction detection (other two baselines are not applicable). To detect interactions, BANN first calculates the interaction functions from eq.(5.3), then their empirical ℓ_2 norms are used as the "interaction strength", and then BANN selects the top k interactions. Possible interaction candidates are based on the group-Lasso clusters for every "sub-NN" in our additive model. For NID, we use the setup in Tsang et al. (2018). We run the same experiments as the RMSE setting using Tab. 5.1. To assess ranking quality, we use the top-rank recall metric Tsang et al. (2018): a recall of interaction rankings where only those interactions that are correctly ranked before we encounter any false positives are considered. Only one superset interaction from each sub-function of f is counted as a true interaction. From Tab. 5.1, we see that the BANN outperforms NID for interaction detection.

Four different types of BANN. As described in section 5.2, we can derive other deep probabilistic model schemes for BANN: MC dropout, random features (RF), deep kernel learning (DKL) and deep random features (DRF), see section 5.2. Then, we can calculate uncertainty based on each of these schemes. Here, we consider two measures, the mean log likelihood (MLL) and the empirical ℓ_2 norm of the interaction or main effect function from eq.(5.3). The mean log likelihood (MLL) measures the prediction accuracy. The empirical ℓ_2 norm measures the strength of the interaction or main effect. For the uncertainty of the interaction measure, we only calculate it for BANN but do not compare the results with NID Tsang et al. (2018) since NID cannot model uncertainty. In Tab. 5.2, we show the two types

measures with uncertainties for f_1 in Tab. 5.1. The results show that all four types of constructions (derivable from our proposal) correctly yields interactions between x_1 and x_2 as well as the main effects of each input because the values are close to the truth from the math calculation. We also see the advantage of our framework that it makes the comparison between different deep probabilistic models easy for new computation skeletons.

Interpretability and uncertainty using BANN. We showed that BANN can produce interaction effect measure with it's uncertainty. Here, we show that the BANN can also produce uncertainties for the interaction function defined in eq. (5.3). We use the f_1 in Tab. 5.1 as an example and we are interested in modeling the interaction between x_1 and x_2. Using BANN, we can obtain the interaction function and the uncertainty function for it in Fig. 5.2. We draw them as heatmaps where the horizontal axis is $\mathbf{x}_1$ and the vertical axis is $\mathbf{x}_2$. We can see from those figures that BANN can provide meaningful information for interactions, which is rarely available for deep neural network models. This ability of BANN can be very useful for interpretability.

Benchmark experiments. Finally, we apply BANN on common datasets used by other authors. BANN (which is a more compact model) yields competitive performance in addition to the other features it natively provides such as interaction (interpretability) and uncertainty discussed above. This implies that these additional benefits do not come at a cost of performance.

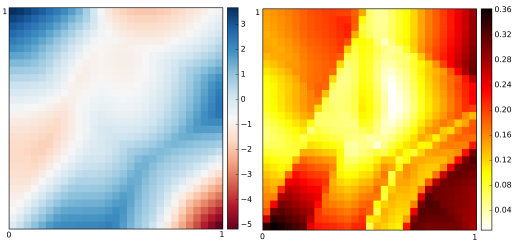


Figure 5.2: Interaction between x_1, x_2 for f_1 in Tab. 5.1. Mean interaction (L) and its standard deviation (R) shown.

Table 5.3: Average test performance in RSME and Standard Errors for BANN(ours), dropout uncertainty (MC dropout), deep Gaussian process (DGP 5) and probabilistic back-propagation(PBP) on benchmarks. Dataset size(N) and input dimensionality(Q) are also given.

	N	Q	BANN	MC dropout	DGP 5	PBP
Boston	506	13	3.03 ± 0.12	2.97 ± 0.19	2.92 ± 0.17	3.01 ± 0.18
Concrete	1030	8	5.18 ± 0.14	5.23 ± 0.12	5.65 ± 0.10	5.67 ± 0.09
Energy	768	8	0.65 ± 0.03	1.66 ± 0.04	0.47 ± 0.01	1.80 ± 0.05
Kin8nm	8192	8	0.07 ± 0.00	0.10 ± 0.00	0.06 ± 0.00	0.10 ± 0.00
Naval	11934	16	0.01 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.01 ± 0.00
Power	9568	4	4.04 ± 0.03	4.02 ± 0.04	3.68 ± 0.03	4.12 ± 0.03
Protein	45730	9	4.07 ± 0.01	4.36 ± 0.01	3.72 ± 0.04	4.73 ± 0.01
Wine	1599	11	0.66 ± 0.01	0.62 ± 0.01	0.63 ± 0.01	0.64 ± 0.01

5.5 Discussion

We presented a framework to understand the flow of Information in deep probabilistic models. We generalized the structure on function class assumption in statistics to deep probabilistic models through the use of the computation skeleton, which also helps provide more interpretable models. Our framework ties various deep probabilistic models together including BNNs, DGPs, DKL and others. As a by-product, our technical development and analysis helps us understand the similarities and difference between various deep probabilistic models. Finally, we show how the structure assumption can be instantiated via an additive structure assumption to derive a Bayesian additive neural networks (BANN), which produce interpretable results through statistical interactions with uncertainty estimates.

6 CONCLUDING REMARKS

In this dissertation, we discussed how kernel methods can be used in the analysis of big and complex data. In Chapter 2, we considered the situation when multiple datasets are available and come from heterogeneous sources. Those datasets formulate a large dataset with more samples but also with more biases. Our contribution is to derive a framework using the graphical causal model to identify covariate shift bias and using maximum mean discrepancy, a kernel-based approach, to eliminate the covariate shift bias between multiple datasets. The framework is applied to Alzheimer's disease study to show an advantage. More details can be found in Zhou et al. (2016, 2018a).

When we have multiple datasets from heterogeneous sources, we can use the framework in Chapter 2 to eliminate the biases with the best effort. However, an important question remains to be answered that how we can determine combining multiple datasets is beneficial because of increased sample size instead of harming the analysis because of increased biases. In Chapter 3, we answered this question. We derived a hypothesis testing to decide when it is beneficial to combine multiple datasets from heterogeneous sources. We derived theoretical analysis for a sparse multi-source model on multi-source datasets to study its behavior. More details can be found in Zhou et al. (2017).

In the analysis of spatiotemporal data, we are interested in understanding the dynamic behavior. For example, in Chicago crime data, we want to understand how crime happens and transit between various communities in Chicago as time changes. Instead of fitting a linear auto-regressive model, we built a kernel-based non-parametric model to study this dynamic behavior. In this setting, the spatial information and kernel space formulates a high dimensional space while the samples are sparse in the observed period. As prior high dimensional statistics, we added regularizations to solve the problem in Chapter 4. We derived theoretical analysis to study the behavior of the non-parametric kernel-based model in this setting when model space is high dimension and samples are dependent. More

details can be found in Zhou and Raskutti (2019).

Finally, with the development of high computation power and the ability to collect big data, deep learning becomes popular in a lot of applications recently. The kernel methods can not directly adapt to this change and benefit from the recent high computation power. We found a new area called deep probabilistic models. In Chapter 5, we derived a framework and showed that kernels and structure assumptions in statistics can be used to understand those models. It also points out a way to extend classic statistics models, such as additive model and hierarchical model, to a deep structure where we can use the recent development from deep learning and maintain interpretability and uncertainty estimates from statistics. More details can be found in Zhou et al. (2018b).

A APPENDIX

A.1 Proofs for Theorems in Chapter 2

In this section, we give proofs for Theorem 2.2, Theorem 2.6, Theorem 2.7, and Theorem 2.9.

Proof for Theorem 2.2

Theorem A.1. *The distribution shift correction is identifiable if there exists a known set of variables Z such that the following three conditions are all concurrently satisfied:*

1) *Z d-separates X and E_B (sample selection bias) and also d-separates X and E_P (population characteristic difference);*

2) *The conditional probability $\mathbb{P}(X|Z)$, after appropriate transformations on X, is the same across multiple participating sites (S and T);*

3) *The distribution of Z has a non-trivial overlap across multiple sites (S and T), which means that there exists an interval $[a, b]$ such that $\mathbb{P}(a \leqslant Z \leqslant b) \geqslant 0.5$ for all sites.*

Proof. We denote measurements of interest as X_S in dataset 1 and X_T in dataset 2. We assume $E_B = 1$, $E_P = 1$ for dataset 1 and $E_B = 2$, $E_P = 2$ for dataset 2 to represent the biases between the two datasets. Without loss of generality, we assume that a transformation $h^{\lambda_0}(.)$ can resolve the distributional shift, that is, marginal distributions of $h^{\lambda_0}(X_S)$ and X_T are the same. However, when biases exist, we have,

$$\mathbb{P}(h^{\lambda_0}(X_S)|E_B = 1, E_P = 1) \neq P(X_T|E_B = 2, E_P = 2). \tag{A.1}$$

Therefore, we may not be able to find the correct transformation by matching the distributions Zhou et al. (2016). Then, the correction problem becomes non-identifiable if we do not have any additional information. The situation changes when we have a set of variables Z which satisfy the three identification conditions

in Thm. 2.2. The following explanation describes why.

$$\mathbb{P}(h^{\lambda_0}(X_S)|E_B = 1, E_P = 1) = \mathbb{E}_{Z|E_B=1,E_P=1}[\mathbb{P}(h^{\lambda_0}(X_S)|Z, E_B = 1, E_P = 1)] \quad (A.2)$$

$$= \mathbb{E}_{Z|E_B=1,E_P=1}[\mathbb{P}(h^{\lambda_0}(X_S)|Z)] \quad (A.3)$$

The second equation holds because of condition 1) in Thm. 2.2. Similarly,we have,

$$\mathbb{P}(X_T|E_B = 2, E_P = 2) = \mathbb{E}_{Z|E_B=2,E_P=2}[\mathbb{P}(X_T|Z)] \quad (A.4)$$

Then, since condition 2) holds, we have $\mathbb{P}(h^{\lambda_0}(X_S)|Z) = \mathbb{P}(X_T|Z)$, that is, they are identical functions of Z. Therefore, the only difference between the two datasets are $\mathbb{P}(Z|E_B = 1, E_P = 1)$ and $\mathbb{P}(Z|E_B = 2, E_P = 2)$. We should keep in mind that this difference is similar to [A.1]. However, there is no longer an unknown transformation $h^{\lambda_0}(.)$ in the relation. To address this issue, we can conduct a subsampling procedure $\mathcal{SSP}$ on Z to approximately align $\mathbb{P}(Z|E_B = 1, E_P = 1)$ with $\mathbb{P}(Z|E_B = 2, E_P = 2)$. The condition 3) in Thm. 2.2 shows that this is possible. After the subsampling procedure, we will approximately have,

$$\mathbb{P}(h^{\lambda_0}(X_S)|E_B = 1, E_P = 1, \mathcal{SSP}) = \mathbb{P}(X_T|E_B = 2, E_P = 2, \mathcal{SSP}). \quad (A.5)$$

Therefore, the correction problem becomes identifiable since we can now learn h^{λ_0} by matching the distributions. □

Proofs for Theorem 2.6 and Theorem 2.7

First,we have Lemmas from Zhou et al. (2016).

Lemma A.2. *For any fixed function* $h(x_s, \lambda)$, $g(x_t, \beta)$,*any* λ, β, *bounded kernel* $\mathcal{K}$, *we*

have

$$P(\sup_{f\in F}|\frac{1}{n}\sum_{i=1}^{n} f(h(x_s^i,\lambda)) - E_{x_s} f(h(x_s,\lambda))| - E_{x_s}\sup_{f\in F}|\frac{1}{n}\sum_{i=1}^{n} f(h(x_s^i,\lambda)) - E_{x_s} f(h(x_s,\lambda))| > \epsilon)$$
$$\leqslant exp(-\frac{\epsilon^2 n}{2K})$$
$$P(\sup_{f\in F}|\frac{1}{m}\sum_{i=1}^{m} f(g(x_t^i,\beta)) - E_{x_t} f(g(x_t,\beta))| - E_{x_t}\sup_{f\in F}|\frac{1}{m}\sum_{i=1}^{m} f(g(x_t^i,\beta)) - E_{x_t} f(g(x_t,\beta))| > \epsilon)$$
$$\leqslant exp(-\frac{\epsilon^2 m}{2K})$$

Lemma A.3. *For any fixed function* $h(x_s,\lambda)$, $g(x_t,\beta)$,*any* λ, β, *bounded kernel* $\mathcal{K}$, *we have*

$$E_{x_s}\sup_{f\in F}|\frac{1}{n}\sum_{i=1}^{n} f(h(x_s^i,\lambda)) - E_{x_s} f(h(x_s,\lambda))| \leqslant \frac{2\sqrt{K}}{\sqrt{n}}$$
$$E_{x_t}\sup_{f\in F}|\frac{1}{m}\sum_{i=1}^{m} f(g(x_t^i,\beta)) - E_{x_t} f(g(x_t,\beta))| \leqslant \frac{2\sqrt{K}}{\sqrt{m}}$$

Lemma A.4. *For any fixed function* $h(x_s,\lambda)$, $g(x_t,\beta)$ *and a bounded kernel* $\mathcal{K}$, *if (A1) holds, we have*

$$P(\sup_{\lambda\in\Omega_\lambda}\sup_{f\in F}|\frac{1}{n}\sum_{i=1}^{n} f(h(x_s^i,\lambda)) - E_{x_s} f(h(x_s,\lambda))| > \frac{\sqrt{K}}{\sqrt{n}}\left(4+\sqrt{C^{(h,\alpha)}+\frac{d_\lambda}{2r_h}\log n}\right)) \leqslant \frac{\alpha}{2}$$
$$P(\sup_{\beta\in\Omega_\beta}\sup_{f\in F}|\frac{1}{m}\sum_{i=1}^{m} f(g(x_t^i,\beta)) - E_{x_t} f(g(x_t,\beta))| > \frac{\sqrt{K}}{\sqrt{m}}\left(4+\sqrt{C^{(g,\alpha)}+\frac{d_\beta}{2r_g}\log m}\right)) \leqslant \frac{\alpha}{2}$$

where $C^{(h,\alpha)} = \log(2|\Omega_\lambda|)+\log\alpha^{-1}+\frac{d_\lambda}{r_h}\log\frac{L_h}{\sqrt{K}}$, *and* $C^{(g,\alpha)} = \log(2|\Omega_\beta|)+\log\alpha^{-1}+\frac{d_\beta}{r_g}\log\frac{L_g}{\sqrt{K}}$

Theorem A.5. *Under mild assumptions 2.4, if there is a* λ_0,θ_0 *such that* $h^{\lambda_0}(X_S)$ *and* $g^{\theta_0}(X_T)$ *have the same distribution, then*

$$\mathcal{MMD}(h^{\hat{\lambda}}(X_S), g^{\hat{\theta}}(X_T)) \to 0$$

with the rate $\max(\frac{\sqrt{\log(n_S)}}{\sqrt{n_S}}, \frac{\sqrt{\log(n_T)}}{\sqrt{n_T}})$. *If* λ_0, θ_0 *are unique, then the estimators* $\hat{\lambda}, \hat{\theta}$ *are consistent.*

Proof. Recall the basic inequality

$$\sup_{f \in F}(\frac{1}{m}\sum_{i=1}^{m} f(g(x_t^i, \hat{\beta})) - \frac{1}{n}\sum_{i=1}^{n} f(h(x_s^i, \hat{\lambda}))) \leqslant \sup_{f \in F}(\frac{1}{m}\sum_{i=1}^{m} f(g(x_t^i, \beta_0)) - \frac{1}{n}\sum_{i=1}^{n} f(h(x_t^i, \lambda_0)))$$

$$\|E_{x_s^i}\mathcal{K}(h(x_s^i,\hat{\lambda}),.) - E_{x_t}\mathcal{K}(g(x_t,\hat{\beta})\|_H - \|E_{x_s^i}\mathcal{K}(h(x_s^i,\lambda_0),.) - E_{x_t}\mathcal{K}(g(x_t,\beta_0)\|_H$$
$$= \sup_{f\in F}(E_{x_t}f(g(x_t,\hat{\beta})) - E_{x_s^i}f(h(x_s^i,\hat{\lambda}))) - \sup_{f\in F}(E_{x_t}f(g(x_t,\beta_0)) - E_{x_s^i}f(h(x_s^i,\lambda_0)))$$

We use a basic inequality and get

$$\leqslant \sup_{f\in F}(E_{x_t}f(g(x_t,\hat{\beta})) - E_{x_s^i}f(h(x_s^i,\hat{\lambda}))) - \sup_{f\in F}(E_{x_t}f(g(x_t,\beta_0)) - E_{x_s^i}f(h(x_s^i,\lambda_0)))$$
$$+ \sup_{f\in F}(\frac{1}{m}\sum_{i=1}^{m} f(g(x_t^i,\beta_0)) - \frac{1}{n}\sum_{i=1}^{n} f(h(x_t^i,\lambda_0))) - \sup_{f\in F}(\frac{1}{m}\sum_{i=1}^{m} f(g(x_t^i,\hat{\beta})) - \frac{1}{n}\sum_{i=1}^{n} f(h(x_s^i,\hat{\lambda})))$$
$$\leqslant |\sup_{f\in F}(\frac{1}{m}\sum_{i=1}^{m} f(g(x_t^i,\hat{\beta})) - \frac{1}{n}\sum_{i=1}^{n} f(h(x_s^i,\hat{\lambda}))) - \sup_{f\in F}(E_{x_t}f(g(x_t,\hat{\beta})) - E_{x_s^i}f(h(x_s^i,\hat{\lambda})))|$$
$$+ |\sup_{f\in F}(\frac{1}{m}\sum_{i=1}^{m} f(g(x_t^i,\beta_0)) - \frac{1}{n}\sum_{i=1}^{n} f(h(x_s^i,\lambda_0))) - \sup_{f\in F}(E_{x_t}f(g(x_t,\beta_0)) - E_{x_s^i}f(h(x_s^i,\lambda_0)))|$$

We use the fact that $|\sup_x f_1(x) - \sup_x f_2(x)|$ is smaller than $\sup_x |f_1(x) - f_2(x)|$

$\hat{\lambda},\lambda_0 \in \Omega_\lambda$, $\hat{\beta},\beta_0 \in \Omega_\beta$ and get

$$\leqslant 2 \sup_{\lambda\in\Omega_\lambda,\beta\in\Omega_\beta} \sup_{f\in F} |(\frac{1}{m}\sum_{i=1}^{m} f(g(x_t^i,\beta)) - \frac{1}{n}\sum_{i=1}^{n} f(h(x_s^i,\lambda))) - (E_{x_t}f(g(x_t,\beta)) - E_{x_s^i}f(h(x_s^i,\lambda)))|$$
$$\leqslant 2 \sup_{\lambda\in\Omega_\lambda,\beta\in\Omega_\beta} \sup_{f\in F} |(\frac{1}{m}\sum_{i=1}^{m} f(g(x_t^i,\beta)) - E_{x_t}f(g(x_t,\beta))|$$
$$+ 2 \sup_{\lambda\in\Omega_\lambda,\beta\in\Omega_\beta} \sup_{f\in F} |(\frac{1}{n}\sum_{i=1}^{n} f(h(x_s^i,\lambda)) - E_{x_s^i}f(h(x_s^i,\lambda)))|$$
$$= 2 \sup_{\beta\in\Omega_\beta} \sup_{f\in F} |(\frac{1}{m}\sum_{i=1}^{m} f(g(x_t^i,\beta)) - E_{x_t}f(g(x_t,\beta))|$$
$$+ 2 \sup_{\lambda\in\Omega_\lambda} \sup_{f\in F} |(\frac{1}{n}\sum_{i=1}^{n} f(h(x_s^i,\lambda)) - E_{x_s^i}f(h(x_s^i,\lambda)))|$$

The results follows from Lemma A.4, by noticing that for every random variable W, Z, constant a, b, we have $P(W + Z > a + b) \leqslant P(W > a) + P(Z > b)$. Thus, for

any $\alpha > 0$, with probability at least $1 - \alpha$.

$$\begin{aligned}&\|E_{x_s^i}\mathcal{K}(h(x_s^i,\hat{\lambda}),.) - E_{x_t}\mathcal{K}(g(x_t,\hat{\beta})\|_H - \|E_{x_s^i}\mathcal{K}(h(x_s^i,\lambda_0),.) - E_{x_t}\mathcal{K}(g(x_t,\beta_0)\|_H \\ &\leqslant 2\frac{\sqrt{K}}{\sqrt{n}}\left(4+\sqrt{C^{(h,\alpha)}+\frac{d_\lambda}{2r_h}\log n}\right) + 2\frac{\sqrt{K}}{\sqrt{m}}\left(4+\sqrt{C^{(g,\alpha)}+\frac{d_\beta}{2r_g}\log m}\right)\end{aligned}$$

where $C^{(h,\alpha)} = \log(2|\Omega_\lambda|)+\log\alpha^{-1}+\frac{d_\lambda}{r_h}\log\frac{L_h}{\sqrt{K}}$, and $C^{(g,\alpha)} = \log(2|\Omega_\beta|)+\log\alpha^{-1}+\frac{d_\beta}{r_g}\log\frac{L_g}{\sqrt{K}}$

Next, we show the estimators are consistent. We assume that $\Omega_\lambda, \Omega_\beta$ are bounded. For notational convenience we simply call

$$\|E_{x_s}\mathcal{K}(h(x_s,\hat{\lambda}),.) - E_{x_t}\mathcal{K}(g(x_t,\hat{\beta}),.)\|_H - \|E_{x_s}\mathcal{K}(h(x_s,\lambda_0),.) - E_{x_t}\mathcal{K}(g(x_t,\beta_0),.)\|_H$$

as $\zeta(\hat{\lambda},\hat{\beta})$.

Notice that $\zeta(\cdot)$ is continuous because $\zeta(\cdot)^2$ is the summation of expectations of bounded continuous functions (because the kernel is bounded continuous). If $(\hat{\lambda},\hat{\beta})$ doesn't converge to (λ_0,β_0) when $\zeta(\hat{\lambda},\hat{\beta})$ converges to 0, then we have a sequence $(\hat{\lambda}_k,\hat{\beta}_k)$ and an $\epsilon > 0$, such that $\|(\hat{\lambda}_k,\hat{\beta}_k) - (\lambda_0,\beta_0)\| > \epsilon$ but $\zeta(\hat{\lambda}_k,\hat{\beta}_k)$ converges to 0. Because $\Omega_\lambda, \Omega_\beta$ bounded, $\zeta(\cdot)$ is continuous, and hence $T(\lambda,\beta,C) = \{\lambda \in \Omega_\lambda, \beta \in \Omega_\beta | \zeta(\lambda,\beta) < C\}$ is a compact set of (λ,β) for some constant C.

So we can find a point $(\tilde{\lambda},\tilde{\beta})$ in $T(\lambda,\beta,C)\cap\{(\lambda,\beta)|\|(\lambda,\beta)-(\lambda_0,\beta_0)| > \epsilon\}$ such that there is a subsequence $(\hat{\lambda}_{k_l},\hat{\beta}_{k_l})$ which converges to $(\tilde{\lambda},\tilde{\beta})$ when l goes to ∞, based on Bolzano-Weierstrass theorem. But since $\zeta(\cdot)$ is continuous, we have $\zeta(\tilde{\lambda},\tilde{\beta}) = 0$, with $\|(\tilde{\lambda},\tilde{\beta}) - (\lambda_0,\beta_0)\| > \epsilon$. This contradicts with the unique solution requirement of (λ_0,β_0). □

Theorem A.6. *Given Assumptions 2.4, when* H_0 *is true, with probability at least* $1-\alpha$,

$$\widehat{\mathcal{MMD}}(h^{\hat{\lambda}}(X_S), g^{\hat{\theta}}(X_T)) \leqslant \sqrt{\frac{2k(n_S+n_T)\log\alpha^{-1}}{n_S n_T}} + 2\sqrt{\frac{k}{n_S}} + 2\sqrt{\frac{k}{n_T}}$$

When H_A *is true, with probability at least* $1-\alpha$*,*

$$|\widehat{\mathcal{MMD}}(h^{\hat{\lambda}}(X_S), g^{\hat{\theta}}(X_T)) - C_*| \leqslant \sqrt{\frac{k}{n_S}}\left(4+\sqrt{C^{h,\alpha}+\frac{d_\lambda}{2r_h}\log n_S}\right)+\sqrt{\frac{k}{n_T}}\left(4+\sqrt{C^{g,\alpha}+\frac{d_\theta}{2r_g}\log n_T}\right)$$

where $C_* = \min_{\lambda,\theta}\mathcal{MMD}(h^\lambda(P_S), g^\theta(P_T))$ *is a positive constant when* H_A *holds. Here,* $C^{h,\alpha} = \log(2|\Omega_\lambda|)+\log\alpha^{-1}+\frac{d_\lambda}{r_h}\log\frac{L_h}{\sqrt{k}}$ *and* $C^{g,\theta} = \log(2|\Omega_\theta|)+\log\alpha^{-1}+\frac{d_\theta}{r_g}\log\frac{L_g}{\sqrt{k}}$

Proof. Under $\mathbf{H_0}$,

$$\begin{aligned}
&\mathcal{M}(\hat{\lambda},\hat{\beta}) - \mathcal{M}^*(\lambda_0,\beta_0)\\
&\leqslant \mathcal{M}(\lambda_0,\beta_0) - \mathcal{M}^*(\lambda_0,\beta_0)\\
&= \mathrm{MMD}(h(x_s,\lambda_0), g(x_t,\beta_0)) - \mathrm{MMD}^*(h(x_s,\lambda_0), g(x_t,\beta_0))
\end{aligned}$$

where $\mathrm{MMD}^*(\cdot)$ is the MMD in the population sense while $\mathrm{MMD}(\cdot)$ takes the expectation in a sample sense. The MMD empirical bound from Theorem 7 in Gretton et al. (2012) can be directly applied to the right hand side of the above inequality. This application will lead to the bound on $\mathbf{H_0}$.

Similarly, under $\mathbf{H_A}$,

$$\begin{aligned}
&\mathcal{M}(\hat{\lambda},\hat{\beta}) - \mathcal{M}^*(\lambda_A,\beta_A)\\
&\leqslant \mathcal{M}(\lambda_A,\beta_A) - \mathcal{M}^*(\lambda_A,\beta_A)\\
&= \mathrm{MMD}(h(x_s,\lambda_A), g(x_t,\beta_A)) - \mathrm{MMD}^*(h(x_s,\lambda_A), g(x_t,\beta_A))
\end{aligned}$$

Similar to the case of $\mathbf{H_0}$, the upper bound follows from Theorem 7 of Gretton et al. (2012). The lower bound proof under the alternative follows from Lemma A.4.

$$|\mathcal{M}(\hat{\lambda},\hat{\beta})-\mathcal{M}^*(\lambda_A,\beta_A)| = |\min_{\lambda\in\Omega_\lambda}\min_{\beta\in\Omega_\beta}\sup_{f\in F}(\frac{1}{m}\sum_{i=1}^{m}f(g(x_t^i,\beta))-\frac{1}{n}\sum_{i=1}^{n}f(h(x_s^i,\lambda)))$$
$$-\min_{\lambda\in\Omega_\lambda}\min_{\beta\in\Omega_\beta}\sup_{f\in F}(E_{x_t}f(g(x_t,\beta))-E_{x_s}f(h(x_s,\lambda)))|$$

Use the fact that $|\min_x f_1(x)-\min_x f_2(x)|$ is smaller than $\sup_x|f_1(x)-f_2(x)|$, we have

$$|\mathcal{M}(\hat{\lambda},\hat{\beta})-\mathcal{M}^*(\lambda_A,\beta_A)|$$
$$\leqslant \sup_{\lambda\in\Omega_\lambda,\beta\in\Omega_\beta}\sup_{f\in F}|(\frac{1}{m}\sum_{i=1}^{m}f(g(x_t^i,\beta))-\frac{1}{n}\sum_{i=1}^{n}f(h(x_s^i,\lambda)))-(E_{x_t}f(g(x_t,\beta))-E_{x_s}f(h(x_s,\lambda)))|$$
$$\leqslant \sup_{\beta\in\Omega_\beta}\sup_{f\in F}|\frac{1}{m}\sum_{i=1}^{m}f(g(x_t^i,\beta))-E_{x_t}f(g(x_t,\beta))|+\sup_{\lambda\in\Omega_\lambda}\sup_{f\in F}|\frac{1}{n}\sum_{i=1}^{n}f(h(x_s^i,\lambda))-E_{x_s}f(h(x_s,\lambda))|$$

The results come from Lemma A.4, and the fact that for every random variable W, Z, constant a, b, we have $P(W+Z>a+b)\leqslant P(W>a)+P(Z>b)$ □

Proof for Theorem 2.9

Theorem A.7. *Define* $g^b_{u(i,k)}$ *to be the number of appearances of* $x_u^{(i,k)}$ *in iteration* b. *Define* $\mathbb{COV}(g_{u(i,k)},\lambda)=\frac{1}{B}\sum_{b=1}^{B}(\hat{\lambda}^b-\hat{\lambda})(g^b_{u(i,k)}-\frac{s_i}{n_u^i})$. *The IJ estimator of variance for* $\hat{\lambda}$ *is*

$$\mathbb{VAR}_{IJ}(\hat{\lambda})=\sum_{u\in\{S,T\}}\sum_{i=1}^{d}\sum_{k=1}^{n_u^i}(\mathbb{COV}(g_{u(i,k)},\lambda))^2$$

The procedure for $\hat{\theta}$ *is identical.*

We provide a proof of our extension of the infinitesimal Jackknife estimator Efron (2014) to multiple groups. The proofs proceed in two steps.

Proof. **Step 1.**

Before describing the proof, we define some notations. We have two datasets X_S, X_T and d groups for each dataset. We define our samples from one dataset as

$$X_S = (x_S^{(1,1)}, ..., x_S^{(1,n_S^1)}, ..., x_S^{(d,1)}, ..., x_S^{(d,n_S^d)})$$

and samples from the other dataset as

$$X_T = (x_T^{(1,1)}, ..., x_T^{(1,n_T^1)}, ..., x_T^{(d,1)}, ..., x_T^{(d,n_T^d)}).$$

Based on $X = (X_S, X_T)$, we can generate bootstrap samples for every group and we call all such generated samples as $V = (V_S, V_T)$. In this proof (similar to Efron (2014)), we assume that the bootstrap sample sizes are the same as the original samples, that is,

$$V_S = (v_S^{(1,1)}, ..., v_S^{(1,n_S^1)}, ..., v_S^{(d,1)}, ..., v_S^{(d,n_S^d)})$$

and

$$V_T = (v_T^{(1,1)}, ..., v_T^{(1,n_T^1)}, ..., v_T^{(d,1)}, ..., v_T^{(d,n_T^d)}).$$

Based on $V = (V_S, V_T)$, we define the count variables $\mathcal{N}(V) = (\mathcal{N}(V_S), \mathcal{N}(V_T))$ and probability variables $\mathcal{P}(V) = (\mathcal{P}(V_S), \mathcal{P}(V_T))$, which are,

$$\mathcal{N}(V_u)^{(i,j)} = \#\{k = 1, ..., n_u^i | V_u^{(i,k)} = X_u^{(i,j)}\}, \text{ for any } u \in \{S, T\}.\ i \in \{1, ..., d\}.\ j \in \{1, ..., n_u^i\}.$$

$$\mathcal{P}(V_u)^{(i,j)} = \frac{1}{n_u^i}\mathcal{N}(V_u)^{(i,j)}, \text{ for any } u \in \{S, T\}.\ i \in \{1, ..., d\}.\ j \in \{1, ..., n_u^i\}.$$

In other words, $\mathcal{N}(V_u)^{(i,j)}$ records how many times $X_u^{(i,j)}$ appears in the bootstrap samples for one iteration. The probability variable $\mathcal{P}(V_u)^{(i,j)}$ is the normalized $\mathcal{N}(V_u)^{(i,j)}$ such that $\sum_{j=1}^{n_u^i} \mathcal{P}(V_u)^{(i,j)} = 1$. The bootstrap process is repeated B times. Later, when we have superscript b for V, $\mathcal{N}(V)$ and $\mathcal{P}(V)$, it implies that those variables are related to the b^{th} iteration of the bootstrap process. Now, we define another term to be the baseline for the probability variables, which is,

$$\mathcal{P}(X) = (\mathcal{P}(X_S), \mathcal{P}(X_T)), \text{ where } \mathcal{P}(X_u)^{(i,j)} = \frac{1}{n_u^i}, \text{ for any } u \in \{S, T\}.\ i \in \{1, ..., d\}.\ j \in \{1, ..., n_u^i\}.$$

We can see that $\mathcal{P}(X)$ represents the uniform distribution for each group of X, which only depends on the original samples X. On the other hand, $\mathcal{P}(V^b)$ depends on X and the bootstrap samples V^b for the b^{th} iteration.

The probability variables $\mathcal{P}(V^b)$ and the baseline probability variables $\mathcal{P}(X)$ are connected by multinomial distributions as

$$n_u^i \times \mathcal{P}(V_u^b)^{(i)} \sim \mathbf{Multi}_{n_u^i}(n_u^i, \mathcal{P}(X_u)^{(i)}), \text{ for any } u \in \{S, T\}. \, i \in \{1, ..., d\}. \, b \in \{1, ..., B\}, \tag{A.6}$$

where $\mathcal{P}(V_u^b)^{(i)}$ is related to the i^{th} group from dataset V_u^b and $\mathcal{P}(X_u)^{(i)}$ is related to the i^{th} group from dataset X_u. In $\mathbf{Multi}_{n_u^i}(n_u^i, \mathcal{P}(X_u)^{(i)})$, n_u^i is the total number of variables in the i^{th} group whereas $\mathcal{P}(X_u)^{(i)}$ is the uniform distribution for this group. This means that $n_u^i \times \mathcal{P}(V_u^b)^{(i)}$ can be viewed as n_u^i samples generated from the multinomial distribution on n_u^i discrete values with equal probability to be drawn.

Let us consider an estimation function f. For the b^{th} iteration in the bootstrap process, we obtain an estimator $\hat{f}^b$ from samples V^b. We assume that, given X, $\hat{f}^b$ only depends on $\mathcal{P}(V^b)$, which means that we can represent $\hat{f}^b$ by $f(\mathcal{P}(V^b))$. Therefore, we can approximate $\hat{f}^b$ via the tangent hyperplane that goes through $f(\mathcal{P}(X))$ at $\mathcal{P}(X)$, which gives us that

$$f_{TAN}(\mathcal{P}(V^b)) = f(\mathcal{P}(X)) + \langle \mathcal{P}(V^b) - \mathcal{P}(X), \mathcal{U} \rangle \tag{A.7}$$

where $\langle \mathcal{P}(V^b) - \mathcal{P}(X), \mathcal{U} \rangle = \sum_{u=S,T} \sum_{i=1}^{d} \langle \mathcal{P}(V_u^b)^{(i)} - \mathcal{P}(X_u)^{(i)}, \mathcal{U}_u^i \rangle$, and $\mathcal{U}_u^i = (\mathcal{U}_u^{(i,1)}, ..., \mathcal{U}_u^{(i,n_u^i)})$,

$$\mathcal{U}_u^{(i,j)} = \lim_{\epsilon \to 0} \frac{f(\mathcal{P}(X_u)^{(i)} + \epsilon(e_j - \mathcal{P}(X_u)^{(i)})) - f(\mathcal{P}(X)^{(i)})}{\epsilon}. \tag{A.8}$$

where e_j is a vector with all zeros except one on the j^{th} position.

Because our bootstrap samples are drawn independently from every group, we

have that

$$\mathbb{E}[f_{TAN}(\mathcal{P}(V^b)) - f(\mathcal{P}(X))]^2 = \sum_{u=S,T}\sum_{i=1}^{d}\mathbb{E}[\langle \mathcal{P}(V_u^b)^{(i)} - \mathcal{P}(X_u)^{(i)}, \mathcal{U}_u^i\rangle]^2 \tag{A.9}$$

Further, using the multinomial distribution relation between $\mathcal{P}(V^b)$ and $\mathcal{P}(X)$ in [A.6], we get

$$\begin{aligned}
&\mathbb{E}[\langle \mathcal{P}(V_u^b)^{(i)} - \mathcal{P}(X_u)^{(i)}, \mathcal{U}_u^i\rangle]^2 \\
&= (\mathcal{U}_u^i)^t\mathbb{E}[(\mathcal{P}(V_u^b)^{(i)} - \mathcal{P}(X_u)^{(i)})(\mathcal{P}(V_u^b)^{(i)} - \mathcal{P}(X_u)^{(i)})^t]\mathcal{U}_u^i \\
&= \frac{1}{{n_u^i}^2}(\mathcal{U}_u^i)^t(I - \frac{1}{n_u^i}1t(1))\mathcal{U}_u^i = \sum_{j=1}^{n_u^i}\frac{1}{{n_u^i}^2}(\mathcal{U}_u^{(i,j)})^2 - \frac{1}{{n_u^i}^3}(\sum_{j=1}^{n_u^i}\mathcal{U}_u^{(i,j)})^2,
\end{aligned} \tag{A.10}$$

where $(\cdot)^t$ represents the transpose operation on a vector or matrix, and 1 is the all one vector. Further, since $\sum_{j=1}^{n_u^i}(e_j - \mathcal{P}(X_u)^{(i)}) = 0$, we have $\sum_{j=1}^{n_u^i}\mathcal{U}_u^{(i,j)} = 0$ from its definition in [A.8]. Therefore, we get an approximation for the mean squared error (MSE) of $\hat{f}^b$ for estimation $f(\mathcal{P}(X))$, that is

$$\mathbb{E}[f_{TAN}(\mathcal{P}(V^b)) - f(\mathcal{P}(X))]^2 = \sum_{u=S,T}\sum_{i=1}^{d}\frac{1}{{n_u^i}^2}\sum_{j=1}^{n_u^i}(\mathcal{U}_u^{(i,j)})^2 \tag{A.11}$$

Further, if $\mathbb{E}[f(\mathcal{P}(V^b))] = f(\mathcal{P}(X))$, then this MSE is the bootstrap estimation for the variance of $f(\mathcal{P}(X))$ Efron (2014).

Step 2.

Now, we consider the estimation regarding the variance of $\hat{\lambda} := \frac{1}{B}\sum_{b=1}^{B}\hat{\lambda}^b$. In this proof, we assume that B is large enough that it exactly covers all finite possibilities for the bootstrap samples. Therefore, we can consider $\hat{\lambda}$ as the value of a function $f(\cdot)$ at $\mathcal{P}(X)$. In other words, we consider $f(\mathcal{P}(X))$ in Step 1 to be $\hat{\lambda}$. Let us imagine that we generate bootstrap samples to estimate the variance of $\hat{\lambda}$ as in Step 1, which is a second layer of bootstrap since $\hat{\lambda}$ is already based on bootstrap samples. Again, we can use the tangent hyperplane scheme in Step 1 to approximate the bootstrap estimation of the variance. For this estimator, the relation $\mathbb{E}[f(\mathcal{P}(V^b))] = f(\mathcal{P}(X)) =$

$\hat{\lambda}$ holds because we assume that B covers all the possibilities. Therefore, the only remaining issue is to calculate $\mathcal{U}_u^{(i,j)}$. We define a probability measure on X which is

$$\begin{aligned}\mathcal{P}(V_u)_\epsilon^{(i,j)} = \{&\mathcal{P}(X_u)^{(i)} + \epsilon(e_j - P(X_u)^{(i)}) \text{ for the group } X_u^{(i)}, \\ &\text{and } \mathcal{P}(X_v)^{(k)} \text{ for other groups } X_v^{(k)}, \text{ for any } v \in \{S, T\}.\ k \in \{1, 2, ..., d\}.\}\end{aligned}$$

Because B bootstrap samples covers all possibilities, we have that

$$f(\mathcal{P}(V_u)_\epsilon^{(i,j)}) = \frac{1}{B}\sum_{b=1}^{B} \hat{\lambda}^b w^b(\mathcal{P}(V_u)_\epsilon^{(i,j)}) \tag{A.12}$$

where $w^b(\mathcal{P}(V_u)_\epsilon^{(i,j)}) = \frac{\mathbb{P}(V^b|\mathcal{P}(V_u)_\epsilon^{(i,j)})}{\mathbb{P}(V^b|\mathcal{P}(X))}$. The probability $\mathbb{P}(V^b|\mu)$ is defined to be the probability of V^b under the probability measure μ. We can further simplify $w^b(\mathcal{P}(V_u)_\epsilon^{(i,j)})$ by

$$\begin{aligned} w^b(\mathcal{P}(V_u)_\epsilon^{(i,j)}) &= \frac{\mathbb{P}(V^b|\mathcal{P}(V_u)_\epsilon^{(i,j)})}{\mathbb{P}(V^b|\mathcal{P}(X))} \\ &= (1 + (n_u^i - 1)\epsilon)^{n_u^i \mathcal{P}(V_u^b)^{(i,j)}} \prod_{l=1, l\neq j}^{n_u^i} (1-\epsilon)^{n_u^i \mathcal{P}(V_u^b)^{(i,l)}} \\ &= (1 + (n_u^i - 1)\epsilon)^{n_u^i \mathcal{P}(V_u^b)^{(i,j)}} (1-\epsilon)^{n_u^i - n_u^i \mathcal{P}(V_u^b)^{(i,j)}} \end{aligned}$$

Let $\epsilon \to 0$, we have that

$$\begin{aligned} &[w^b(\mathcal{P}(V_u)_\epsilon^{(i,j)}) - 1]/\epsilon = [(1 + (n_u^i - 1)\epsilon)^{n_u^i \mathcal{P}(V_u^b)^{(i,j)}} (1-\epsilon)^{n_u^i - n_u^i \mathcal{P}(V_u^b)^{(i,j)}} - 1]/\epsilon \\ &\to [(1 + n_u^i \epsilon n_u^i \mathcal{P}(V_u^b)^{(i,j)})(1 - \epsilon n_u^i) - 1]/\epsilon \\ &\to n_u^i (n_u^i \mathcal{P}(V_u^b)^{(i,j)} - 1) \end{aligned} \tag{A.13}$$

Therefore, using [A.8], we know that

$$\frac{1}{n_u^i}\mathcal{U}_u^{(i,j)} = \frac{1}{B}\sum_{b=1}^{B} \hat{\lambda}^b (n_u^i \mathcal{P}(V_u^b)^{(i,j)} - 1) \tag{A.14}$$

By plugging in that form in equation [A.11], we get the approximation for the bootstrap estimation for the variance of $\hat{\lambda}$, which is

$$\begin{aligned} \mathbb{VAR}(\hat{\lambda}) &= \sum_{u=S,T} \sum_{i=1}^{d} \sum_{j=1}^{n_u^i} (\mathbb{COV}(\hat{\lambda}, \mathcal{N}(V_u)^{(i,j)}))^2, \\ \textit{where } \mathbb{COV}(\hat{\lambda}, \mathcal{N}(V_u)^{(i,j)}) &= \frac{1}{B} \sum_{b=1}^{B} \hat{\lambda}^b ((V_u^b)^{(i,j)} - 1) \end{aligned} \tag{A.15}$$

For finite B, $\frac{1}{B} \sum_{b=1}^{B} \hat{\lambda}^b ((V_u^b)^{(i,j)} - 1)$ is an approximation of $\mathbb{COV}(\hat{\lambda}, \mathcal{N}(V_u)^{(i,j)})$. For the subsampling case, the expectation of $(V_u^b)^{(i,j)}$ is no longer 1 but $\frac{s_i}{n_u^i}$. As a result, we adjust the covariance term to $\mathbb{COV}(\hat{\lambda}, \mathcal{N}(V_u)^{(i,j)}) = \frac{1}{B} \sum_{b=1}^{B} \hat{\lambda}^b ((V_u^b)^{(i,j)} - \frac{s_i}{n_u^i})$ as in Wager and Athey (2017). This leads to the Thm. 2.9 in Chapter 2. □

A.2 Proofs for Theorems in Chapter 3

In this section, we give proofs for Theorem 3.1, Lemma 3.2, Theorem 3.3, Theorem 3.5, Theorem 3.7, and Theorem 3.8 in Chapter 3.

Proofs for Theorem 3.1, Lemma 3.2, Theorem 3.3, and Theorem 3.5

Theorem A.8. $\tau_i = \frac{\sigma_1}{\sigma_i}$ *achieve the smallest variance in* $\hat{\beta}$.

Proof. The choice of τ_i leads to weighted least squares, which is known to be the best linear unbiased estimator (BLUE) under uncorrelated heteroscedastic errors. The variance of $\hat{\beta}$ is equivalent to the case when $\Delta\beta_i = 0$. In the latter case, BLUE condition holds and setting τ_i to the above value achieves lowest variance. The equivalence between variances under two cases completes the proof. □

Lemma A.9. *For model* (3.3)*, we have*

$$\frac{\|\mathrm{Bias}_\beta\|_2^2}{\|G^{-1/2}\Delta\beta\|_2^2} \leqslant \|(\hat{\Sigma}_1^k)^{-2}(\hat{\Sigma}_2^k(n_1\hat{\Sigma}_1)^{-1}\hat{\Sigma}_2^k + \hat{\Sigma}_2^k)\|_*, \tag{A.16}$$

$$\mathrm{Var}_\beta = \sigma_1^2 \left\|(n_1\hat{\Sigma}_1)^{-1} - (n_1\hat{\Sigma}_1 + \hat{\Sigma}_2^k)^{-1}\right\|_*. \tag{A.17}$$

Proof. The estimation from single site model is unbiased, and it has the following variance.

$$\mathrm{Var}_1 = \mathrm{tr}((X_1^T X_1)^{-1})\sigma_1^2 = \mathrm{tr}((n_1\hat{\Sigma}_1)^{-1})\sigma_1^2 \tag{A.18}$$

The estimation error from multi-sites model has the following closed form expres-

sion

$$\hat{\beta} - \beta^* = \left(\begin{pmatrix} X_1 \\ \tau_2 X_2 \\ .. \\ \tau_k X_k \end{pmatrix}^T \begin{pmatrix} X_1 \\ \tau_2 X_2 \\ .. \\ \tau_k X_k \end{pmatrix} \right)^{-1} \begin{pmatrix} \tau_2 X_2 \\ .. \\ \tau_k X_k \end{pmatrix}^T \begin{pmatrix} \tau_2 X_2(\Delta\beta_2) \\ .. \\ \tau_k X_k(\Delta\beta_k) \end{pmatrix}$$
$$+ \left(\begin{pmatrix} X_1 \\ \tau_2 X_2 \\ .. \\ \tau_k X_k \end{pmatrix}^T \begin{pmatrix} X_1 \\ \tau_2 X_2 \\ .. \\ \tau_k X_k \end{pmatrix} \right)^{-1} \begin{pmatrix} X_1 \\ \tau_2 X_2 \\ .. \\ \tau_k X_k \end{pmatrix}^T \begin{pmatrix} \epsilon_1 \\ \tau_2 \epsilon_2 \\ .. \\ \tau_k \epsilon_k \end{pmatrix} \tag{A.19}$$

First term in the summation from (A.19) is bias, while second term is variance. We can see that our choice of $\tau_i = \frac{\sigma_1}{\sigma_i}$ resolves heteroscedastic errors issue among sites. We further simplify bias and variance terms, and obtain

$$Var_2 = tr((n_1\hat{\Sigma}_1 + \sum_{i=2}^{k} n_i \tau_i^2 \hat{\Sigma}_i)^{-1})\sigma_1^2 \tag{A.20}$$

The reduced variance statement is proved. For the bias term, it is equivalent as shown below.

$$\left(\begin{pmatrix} X_1 \\ \tau_2 X_2 \\ .. \\ \tau_k X_k \end{pmatrix}^T \begin{pmatrix} X_1 \\ \tau_2 X_2 \\ .. \\ \tau_k X_k \end{pmatrix} \right)^{-1} \begin{pmatrix} \tau_2 X_2 \\ .. \\ \tau_k X_k \end{pmatrix}^T \begin{pmatrix} \tau_2 X_2 & 0 & ... & 0 \\ 0 & \tau_3 X_3 & ... & 0 \\ 0 & 0 & ... & \tau_k X_k \end{pmatrix}$$
$$\times G^{1/2} \left\{ G^{-1/2} \begin{pmatrix} \Delta\beta_2 \\ .. \\ \Delta\beta_k \end{pmatrix} \right\} \tag{A.21}$$

A one step Cauchy Schwartz inequality is then applied. Then our final proof is to

show $\|..\|_F^2$ on

$$\left(\begin{pmatrix} X_1 \\ \tau_2 X_2 \\ .. \\ \tau_k X_k \end{pmatrix}^T \begin{pmatrix} X_1 \\ \tau_2 X_2 \\ .. \\ \tau_k X_k \end{pmatrix}\right)^{-1} \begin{pmatrix} \tau_2 X_2 \\ .. \\ \tau_k X_k \end{pmatrix}^T \begin{pmatrix} \tau_2 X_2 & 0 & ... & 0 \\ 0 & \tau_3 X_3 & ... & 0 \\ 0 & 0 & ... & \tau_k X_k \end{pmatrix} G^{1/2} \tag{A.22}$$

is equal to right side of the bias relaxation in (3.4).
It is easy to see that $\|A\|_F^2 = \|A^T A\|_*$. Based on this, we can see the first term of matrix inverse contributes the $(\hat{\Sigma}_1^k)^{-2}$ in (3.4). Let the other part in (A.22) be L. We have

$$LL^T = \begin{pmatrix} \tau_2^2 X_2^T X_2 \\ .. \\ \tau_k^2 X_k^T X_k \end{pmatrix}^T G \begin{pmatrix} \tau_2^2 X_2^T X_2 \\ .. \\ \tau_k^2 X_k^T X_k \end{pmatrix} = \begin{pmatrix} n_2 \tau_2^2 \hat{\Sigma}_2 \\ .. \\ n_k \tau_k^2 \hat{\Sigma}_k \end{pmatrix}^T G \begin{pmatrix} n_2 \tau_2^2 \hat{\Sigma}_2 \\ .. \\ n_k \tau_k^2 \hat{\Sigma}_k \end{pmatrix} \tag{A.23}$$

After some manipulations, this becomes $(\hat{\Sigma}_2^k (n_1 \hat{\Sigma}_1)^{-1} \hat{\Sigma}_2^k + \hat{\Sigma}_2^k)$. The bias part is proved. □

Theorem A.10. ***a)*** *Model* (3.3) *has smaller MSE of* $\hat{\beta}$ *than model* (3.1) *whenever*

$$H_0 : \|G^{-1/2} \Delta\beta\|_2^2 \leqslant \sigma_1^2. \tag{A.24}$$

b) *Further, we have the following test statistic,*

$$\left\| \frac{G^{-1/2} \Delta\hat{\beta}}{\sigma_1} \right\|_2^2 \sim \chi^2_{(k-1)*p} \left(\left\| \frac{G^{-1/2} \Delta\beta}{\sigma_1} \right\|_2^2 \right), \tag{A.25}$$

where $\|G^{-1/2} \Delta\beta/\sigma_1\|_2$ *is called a "condition value".*

Proof for Theorem 3.3. **(a):** Based on Lemma 3.2, the theorem is proved when right

side in (3.4) is replaced by

$$\frac{\sigma_1^2 \left\| (n_1\hat{\Sigma}_1)^{-1} - (n_1\hat{\Sigma}_1 + \hat{\Sigma}_2^k)^{-1} \right\|_*}{\|(\hat{\Sigma}_1^k)^{-2}(\hat{\Sigma}_2^k(n_1\hat{\Sigma}_1)^{-1}\hat{\Sigma}_2^k + \hat{\Sigma}_2^k)\|_*} \tag{A.26}$$

We first calculate the numerator

$$\begin{aligned}\sigma_1^2 \left\| (n_1\hat{\Sigma}_1)^{-1} - (n_1\hat{\Sigma}_1 + \hat{\Sigma}_2^k)^{-1} \right\|_* &= \sigma_1^2 \left\| \left[(n_1\hat{\Sigma}_1)^{-1}(n_1\hat{\Sigma}_1 + \hat{\Sigma}_2^k) - I \right] (n_1\hat{\Sigma}_1 + \hat{\Sigma}_2^k)^{-1} \right\|_* \\ &= \sigma_1^2 \left\| (n_1\hat{\Sigma}_1)^{-1}\hat{\Sigma}_2^k(n_1\hat{\Sigma}_1 + \hat{\Sigma}_2^k)^{-1} \right\|_*\end{aligned} \tag{A.27}$$

The denominator is then given by

$$\|(\hat{\Sigma}_1^k)^{-2}(\hat{\Sigma}_2^k(n_1\hat{\Sigma}_1)^{-1}\hat{\Sigma}_2^k + \hat{\Sigma}_2^k)\|_* = \|(\hat{\Sigma}_1^k)^{-2}((\hat{\Sigma}_2^k + n_1\hat{\Sigma}_1)(n_1\hat{\Sigma}_1)^{-1}\hat{\Sigma}_2^k)\|_* \tag{A.28}$$

$$\text{Remember } \hat{\Sigma}_1^k = \hat{\Sigma}_2^k + n_1\hat{\Sigma}_1, \text{, we continue} \tag{A.29}$$

$$= \|((\hat{\Sigma}_2^k + n_1\hat{\Sigma}_1)^{-1}(n_1\hat{\Sigma}_1)^{-1}\hat{\Sigma}_2^k)\|_* = \|((n_1\hat{\Sigma}_1)^{-1}\hat{\Sigma}_2^k(n_1\hat{\Sigma}_1 + \hat{\Sigma}_2^k)^{-1})\|_* \tag{A.30}$$

The last step uses the property of $\|..\|_*$ norm. The proof is completed by noticing the simplified form of numerator and denominator. It is clear now that the right side in (3.4) is exactly σ_1^2.

(b): First, we show $\sigma_1^2 G$ is the covariance matrix of $\Delta\hat{\beta}$. We have

$$cov(\Delta\hat{\beta}_i, \Delta\hat{\beta}_j) = cov(\hat{\beta}_i, \hat{\beta}_j) - cov(\hat{\beta}_i, \hat{\beta}_1) - cov(\hat{\beta}_1, \hat{\beta}_j) + cov(\hat{\beta}_1, \hat{\beta}_1) \tag{A.31}$$

$$\text{Since each site is independent from other site, we have} \tag{A.32}$$

$$cov(\Delta\hat{\beta}_i, \Delta\hat{\beta}_j) = cov(\hat{\beta}_1, \hat{\beta}_1) = \sigma_1^2(n_1\hat{\Sigma}_1)^{-1} \text{for } i \neq j \tag{A.33}$$

$$cov(\Delta\hat{\beta}_i, \Delta\hat{\beta}_i) = cov(\hat{\beta}_i, \hat{\beta}_i) + cov(\hat{\beta}_1, \hat{\beta}_1) \tag{A.34}$$

$$= \sigma_1^2((n_1\hat{\Sigma}_1)^{-1} + (n_i(\sigma_1^2/\sigma_i^2)\hat{\Sigma}_i)^{-1}) = \sigma_1^2((n_1\hat{\Sigma}_1)^{-1} + (n_i\tau_i^2\hat{\Sigma}_i)^{-1}) \tag{A.35}$$

$\Delta\hat{\beta}$ follows Gaussian distribution since it is a linear transformation of Gaussian distribution. It's expectation is $\Delta\beta$ since each $\hat{\beta}_i$ is an unbiased estimator. Hence, we have

$$\Delta\hat{\beta} \sim N(\Delta\beta, \sigma_1^2 G) \tag{A.36}$$

This distribution result, and noticing the connection between Gaussian and non-central χ^2 distributions completes the proof. □

Theorem A.11. *Analysis in Section 3.2 holds for* β *in* (3.9) *by replacing* $\hat{\Sigma}_i$ *with* $\tilde{\Sigma}_i = \hat{\Sigma}_{xx_i} - \hat{\Sigma}_{xz_i}(\hat{\Sigma}_{zz_i})^{-1}\hat{\Sigma}_{zx_i}$

Proof for Theorem 3.5. Define $\gamma^T = (\gamma_1^T, ..., \gamma_k^T)$, $X_{all}^T = \left(X_1^T, \tau_2 X_2^T, ..., \tau_k X_k^T\right)$, $Z_{all} = \mathtt{Diag}\,(Z_1, \tau_2 Z_2, ..., \tau_k Z_k)$. We have

$$\begin{pmatrix} \hat{\beta} \\ \hat{\gamma} \end{pmatrix} - \begin{pmatrix} \beta^* \\ \gamma^* \end{pmatrix} = \begin{pmatrix} X_{all}^T X_{all} & X_{all}^T Z_{all} \\ Z_{all}^T X_{all} & Z_{all}^T Z_{all} \end{pmatrix}^{-1} \begin{pmatrix} X_{all}^T \\ Z_{all}^T \end{pmatrix} \begin{pmatrix} 0 \\ \tau_2 X_2(\Delta\beta_2) \\ .. \\ \tau_k X_k(\Delta\beta_k) \end{pmatrix} + \begin{pmatrix} X_{all}^T X_{all} & X_{all}^T Z_{all} \\ Z_{all}^T X_{all} & Z_{all}^T Z_{all} \end{pmatrix}^{-1} \begin{pmatrix} X_{all}^T \\ Z_{all}^T \end{pmatrix} \begin{pmatrix} \epsilon_1 \\ \tau_2 \epsilon_2 \\ .. \\ \tau_k \epsilon_k \end{pmatrix} \tag{A.37}$$

Using sub-matrix inverse property, we obtain

$$\begin{pmatrix} X_{all}^T X_{all} & X_{all}^T Z_{all} \\ Z_{all}^T X_{all} & Z_{all}^T Z_{all} \end{pmatrix}^{-1} \begin{pmatrix} X_{all}^T \\ Z_{all}^T \end{pmatrix} = \begin{pmatrix} (\tilde{X}_{all}^T \tilde{X}_{all})^{-1} \tilde{X}_{all}^T \\ (\tilde{Z}_{all}^T \tilde{Z}_{all})^{-1} \tilde{Z}_{all}^T \end{pmatrix} \tag{A.38}$$

We then have

$$\tilde{Z}_{all} = (I - X_{all}(X_{all}^T X_{all})^{-1} X_{all}^T) Z_{all} \tag{A.39}$$

$$\tilde{X}_{all} = (I - Z_{all}(Z_{all}^T Z_{all})^{-1} Z_{all}^T) X_{all} = \begin{pmatrix} (I - Z_1(Z_1^T Z_1)^{-1} Z_1^T) X_1 \\ (I - Z_2(Z_2^T Z_2)^{-1} Z_2^T) X_2 \\ .. \\ (I - Z_k(Z_k^T Z_k)^{-1} Z_k^T) X_k \end{pmatrix} \tag{A.40}$$

Define

$$H_{Z_i} = (I - Z_i(Z_i^T Z_i)^{-1} Z_i^T) \tag{A.41}$$

Hence, we have

$$\begin{aligned}\hat{\beta} - \beta^* &= (\tilde{X}_{all}^T \tilde{X}_{all})^{-1} \tilde{X}_{all}^T \begin{pmatrix} 0 \\ \tau_2 X_2(\Delta\beta_2) \\ .. \\ \tau_k X_k(\Delta\beta_k) \end{pmatrix} + (\tilde{X}_{all}^T \tilde{X}_{all})^{-1} \tilde{X}_{all}^T \begin{pmatrix} \epsilon_1 \\ \tau_2 \epsilon_2 \\ .. \\ \tau_k \epsilon_k \end{pmatrix} \\ &= (\tilde{X}_{all}^T \tilde{X}_{all})^{-1} \sum_{i=2}^{k} \tau_i \tilde{X}_i^T X_i(\Delta\beta_i) + (\tilde{X}_{all}^T \tilde{X}_{all})^{-1} \tilde{X}_{all}^T \begin{pmatrix} \epsilon_1 \\ \tau_2 \epsilon_2 \\ .. \\ \tau_k \epsilon_k \end{pmatrix}\end{aligned} \tag{A.42}$$

We also observe that

$$\tilde{X}_i^T X_i = X_i^T H_{Z_i} X_i = X_i^T H_{Z_i}^2 X_i = \tilde{X}_i^T \tilde{X}_i \tag{A.43}$$

Therefore, we can apply our previous results to a subset of parameters if we replace X_i by $\tilde{X}_i$. Since our results only depend on $\hat{\Sigma}_i$, we only need to replace it by

$$\frac{1}{n_i} \tilde{X}_i^T \tilde{X}_i = \frac{1}{n_i} X_i^T H_{Z_i} X_i = \hat{\Sigma}_{x x_i} - \hat{\Sigma}_{x z_i} (\hat{\Sigma}_{z z_i})^{-1} \hat{\Sigma}_{z x_i} \tag{A.44}$$

This proves the theorem. □

Proof of Theorem 3.7:

Theorem A.12. *Let* $0 \leqslant \alpha \leqslant 0.4$. *Assume there exist constants* $0 \leqslant \rho_{min} \leqslant \rho_{max} \leqslant \infty$ *such that*

$$\begin{aligned} &\liminf_{n\to\infty} \phi_{min} \left(s_p \log \bar{n} \left(1 + \frac{2\alpha}{1-2\alpha} \right)^2 \right) \geqslant \rho_{min} \\ &\limsup_{n\to\infty} \phi_{max}(s_p + \min\{\sum_{i=1}^{k} n_i, kp\}) \leqslant \rho_{max}. \end{aligned} \tag{A.45}$$

Then, for $\lambda \propto \sigma\sqrt{\bar{n}\log(kp)}$*, there exists a constant* $\omega > 0$ *such that, with probability converging to 1 for* $n \to \infty$*,*

$$\frac{1}{k}\|\hat{B}^{\lambda} - B^{*}\|_F^2 \leqslant \omega\sigma^2\frac{\bar{s}\log(kp)}{\bar{n}}, \tag{A.46}$$

where $\bar{s} = \{(1-\alpha)\sqrt{s_p} + \alpha\sqrt{s_h/k}\}^2$*,* σ *is the noise level.*

We follow the proof procedure from Lasso Meinshausen and Yu (2009) and group Lasso Liu and Zhang (2009) results. Let B^{λ} be the estimator under the absence of noise, i.e., $B^{\lambda} = \hat{B}^{\lambda,0}$, where $\hat{B}^{\lambda,\xi}$ is defined as in (A.48). The ℓ_2-distance can then be bounded by $\|\hat{B}^{\lambda} - B^{*}\|_F^2 \leqslant 2\|\hat{B}^{\lambda} - B^{\lambda}\|_F^2 + 2\|B^{\lambda} - B^{*}\|_F^2$. The first term on the right-hand side represents the variance of the estimation, while the second term represents the bias. The bias contribution follows directly from Lemma A.13 below, and the variance bound term follows from Lemma A.18.

De-noised response. For $0 < \xi < 1$, we define a de-noised version of the response variable as follows,

$$Y_i(\xi) = X_i\beta_i + \xi\epsilon_i \tag{A.47}$$

We can regulate the amount of noise with the parameter ξ.

For $\xi = 0$, only the signal is retained. The original observations with the full amount of noise are recovered for $\xi = 1$. Now consider for $0 \leqslant \xi \leqslant 1$ the estimator $\hat{B}^{\lambda,\xi}$,

$$\begin{aligned}\hat{B}^{\lambda,\xi} &= \arg\min_{B}\sum_{i=1}^{k}\|Y_i(\xi) - X_i\beta_i\|_2^2 + \lambda\Lambda(B)\\ \Lambda(B) &= (1-\alpha)\sqrt{k}\sum_{j=1}^{p}\|\beta^j\|_2 + \alpha\sum_{j=1}^{p}\|\beta^j\|_1\end{aligned} \tag{A.48}$$

The ordinary sparse multi-site Lasso estimate is recovered under the full amount of noise so that $\hat{B}^{\lambda,1} = \hat{B}^{\lambda}$. Using the notation from the previous results, we have $\hat{B}^{\lambda,0} = B^{\lambda}$, for the estimate in the absence of noise. The definition of the de-noised version of the sparse multi-site Lasso estimator will be helpful for the proof as it

allows to characterize the variance of the estimator.

Part I of proof – Dealing with bias

Let P_* be the set of nonzero groups of B^*, i.e., $P_* = \{j : \beta^j \neq 0\}$. The cardinality of P_* is denoted by s_p. For each j in P_*, let H_j be the set of nonzero elements of β_j, i.e., $H_j = \{i : \beta_i^j \neq 0\}$. The number of all nonzero elements of B is denoted by s_h. For the following, let B^λ be the estimator $\hat{B}^{\lambda,0}$ with no noise (as defined in (A.48)). For each λ, the solution B^λ can be written as $B^\lambda = B^* + \Gamma^\lambda$. We define γ^j and γ_i to be j-th column and i-th row of Γ. γ is the transpose of the unfolded vector of Γ by row. Denote $\lambda_2 = \lambda(1-\alpha)$ and $\eta = \frac{\alpha}{1-\alpha}$. Then

$$\Gamma^\lambda = \arg\min_{\Gamma} f(\Gamma) \tag{A.49}$$

The function $f(\Gamma)$ is given by

$$f(\Gamma) = \bar{n}\gamma^T C\gamma + \lambda_2 \left\{ \sum_{j \in P_*^C} (\sqrt{K}\|\gamma^j\|_2 + \eta\|\gamma^j\|_1) + \sum_{j \in P_*} \sqrt{K}(\|\beta^j + \gamma^j\|_2 - \|\beta^j\|_2) \right\} +$$
$$\lambda_2 \left\{ \sum_{j \in P_*} \eta(\|\beta^j_{H_j} + \gamma^j_{H_j}\|_1 - \|\beta^j H_j)\| + \sum_{j \in P_*} \eta\|\gamma^j H_j^C\|_1 \right\} \tag{A.50}$$

The matrix Γ^λ is the bias of the sparse multi-site Lasso estimator. We derive first a bound on the Frobenius norm of Γ^λ.

Lemma A.13. *Assume conditions in Theorem3.7. The Frobenius norm of* Γ^λ *is then bounded for sufficiently large values of* $\bar{n}$*, given a constant* $\omega_1 > 0$*, by*

$$\|\Gamma^\lambda\|_F^2 \leqslant \omega_1 \sigma^2 \frac{k\bar{s}\log(kp)}{\bar{n}} \tag{A.51}$$

Proof. $f(\Gamma) = 0$ whenever $\Gamma = 0$ following the definition from (A.50). For the true solution Γ^λ, it follows hence that $f(\Gamma^\lambda) \leqslant 0$. For notational simplicity, we drop the

super-script λ from here on. Using $\gamma^T C\gamma \geqslant 0$, we have

$$\left\{ \sum_{j \in P_*^C} (\sqrt{k}\|\gamma^j\|_2) + \sum_{j \in P_*^C} (\eta\|\gamma^j\|_1) + \sum_{j \in P_*} \eta\|\gamma^j_{H_j^C}\|_1 \right\} \leqslant \left\{ \sum_{j \in P_*} \sqrt{k}\|\gamma^j\|_2 + \sum_{j \in P_*} \eta\|\gamma^j_{H_j}\|_1 \right\} \tag{A.52}$$

Since $|P_*| = s_p$, $\sum_{j \in P_*} |H_j| = s_h$. It follows that $\sum_{j \in P_*} \|\gamma^j\|_2 \leqslant \sqrt{s_p}\|\gamma\|_2$, $\sum_{j \in P_*} \|\gamma^j_{H_j}\|_1 \leqslant \sqrt{s_h}\|\gamma\|_2$, and hence, using (A.52),

$$\Lambda(\Gamma) \leqslant 2\{(1-\alpha)\sqrt{ks_p} + \alpha\sqrt{s_h}\}\|\gamma\|_2 = 2\sqrt{k\bar{s}}\|\gamma\|_2 \tag{A.53}$$

Using $f(\Gamma) \leqslant 0$ again and (A.53), it follows that

$$\bar{n}\gamma^T C\gamma \leqslant 2\lambda\sqrt{k\bar{s}}\|\gamma\|_2 \tag{A.54}$$

Now consider $\gamma^T C\gamma$. Bounding this term from below and plugging the result into (A.53) will yield the desired upper bound on the Frobenius norm of Γ. Let $\|\gamma^{(1)}\| \geqslant \|\gamma^{(2)}\| \geqslant ... \geqslant \|\gamma^{(p)}\|$ be the ordered columns of Γ. Let u_n for $n \in N$ be a sequence of positive integers, to be chosen later, and define $U = \{j : \|\gamma^j\|_2 \geqslant \|\gamma^{(u_n)}\|_2\}$. Define $\gamma(U)$ and $\gamma(U^C)$ by setting $\gamma^j(U) = \gamma^j 1\{i \notin U\}$ and $\gamma^j(U^C) = \gamma^j 1\{i \in U\}$, followed by unfolding Γ. Then quantity $\gamma^T C\gamma$ can be written as $\gamma^T C\gamma = \|a + b\|_2^2$, where $a := \bar{n}^{-1/2} X\gamma(U)$, $b := \bar{n}^{-1/2} X\gamma(U^C)$, $X = DIAG(X_1, ..., X_k)$. Then

$$\gamma^T C\gamma = \|a + b\|_2^2 \geqslant (\|a\|_2 - \|b\|_2)^2 \tag{A.55}$$

Before proceeding, we need to bound the norm $\|\gamma(U^C)\|_2$ as a function of u_n. Assume $l = \sum_{j=1}^p \|\gamma^j\|_2$. It holds for every $j = 1, ..., p$ that $\|\gamma^{(j)}\|_2 \leqslant l/j$. Hence,

$$\|\gamma(U^C)\|_2^2 \leqslant (\sum_{j=1}^p \|\gamma^j\|_2)^2 \sum_{j=u_n+1}^p \frac{1}{j^2} \tag{A.56}$$

Therefore, we have

$$\|\gamma(U^C)\|_2 \leqslant \sum_{j=1}^{p} \|\gamma^j\|_2 \sqrt{\frac{1}{u_n}} \leqslant \|\gamma\|_1 \sqrt{\frac{1}{u_n}} \tag{A.57}$$

Based on (A.53), $\Lambda(\Gamma) = (1-\alpha)\sqrt{k}\sum_{j=1}^{p}\|\gamma^j\|_2 + \alpha\|\gamma\|_1$, and (A.57), it follows that

$$\|\gamma(U^C)\|_2^2 \leqslant 4\|\gamma\|_2^2 \left\{ \frac{1}{u_n} \left(\frac{\sqrt{k\bar{s}}}{(1-\alpha)\sqrt{k}+\alpha} \right)^2 \right\} \tag{A.58}$$

By definition, since $\gamma(U)$ has only u_n nonzero groups,

$$\begin{aligned} &\|a\|_2^2 = \|\gamma(U)^T C \gamma(U)\|_2^2 \geqslant \phi_{\min}(u_n)\|\gamma(U)\|_2^2 \geqslant \\ &\phi_{\min}(u_n)\|\gamma\|_2^2 \left(1 - 4\left\{ \frac{1}{u_n} \left(\frac{k\bar{s}}{(1-\alpha)\sqrt{k}+\alpha} \right)^2 \right\}\right) \end{aligned} \tag{A.59}$$

Here we explain why we obtain $\phi_{\min}(u_n)$ instead of $\phi_{\min}(ku_n)$. We denote $\phi^i_{\min}(m)$ to be m-sparse of $\bar{n}^{-1}X_i^T X_i$. Then $\phi_{\min}(m) = \min_{i=1}^{k}\phi^i_{\min}(m)$ because of block structure. Since we have u_n nonzero groups, instead of arbitrary ku_n nonzero elements, we obtain a higher value $\phi_{\min}(u_n) = \min_{i=1}^{k}\phi^i_{\min}(u_n)$ instead of $\phi_{\min}(ku_n)$. This is the one place where we consider the block structure of multi-site design.

As $\gamma(U^C)$ has at most $\min\{\sum_{i=1}^{k} n_i, kp\}$ nonzero groups, using again (A.58), (A.53) and the block structure of multi-site design,

$$\|b\|_2^2 \leqslant 4\phi_{\max}(\min\{\sum_{i=1}^{k} n_i, kp\})\|\gamma\|_2^2 \left\{ \frac{1}{u_n} \left(\frac{\sqrt{k\bar{s}}}{(1-\alpha)\sqrt{k}+\alpha} \right)^2 \right\} \tag{A.60}$$

Using (A.60), (A.59) and (A.55), along with $\phi_{\max}(\min\{\sum_{i=1}^{k} n_i, kp\}) \geqslant \phi_{\min}(u_n)$,

$$\gamma^T C\gamma \geqslant \phi_{\min}(u_n)\|\gamma\|_2^2 \times \left(1 - 4\sqrt{\frac{\phi_{\max}(\min\{\sum_{i=1}^{k} n_i, kp\})}{\phi_{\min}(u_n)} \left\{ \frac{1}{u_n} \left(\frac{\sqrt{k\bar{s}}}{(1-\alpha)\sqrt{k}+\alpha} \right)^2 \right\}}\right) \tag{A.61}$$

Using conditions in Theorem 3.7 and setting $u_n = \log(\bar{n}) \left(\frac{\sqrt{k\bar{s}}}{(1-\alpha)\sqrt{k}+\alpha} \right)^2$, it follows that

$$\gamma^T C\gamma \geqslant \rho_{\min} \left(1 - 4\sqrt{\frac{\rho_{\max}}{\rho_{\min} \log(\bar{n})}} \right) \|\gamma\|_2^2 \tag{A.62}$$

Using this result together with (A.54), which says that $\gamma^T C\gamma \leqslant 2\bar{n}^{-1}\lambda\sqrt{k\bar{s}}\|\gamma\|_2$, we have the following for large $\bar{n}$,

$$\|\Gamma\|_F^2 = \|\gamma\|_2^2 \leqslant \omega_1 \frac{\lambda^2 k\bar{s}}{\bar{n}^2} \tag{A.63}$$

The proof of Lemma A.13 is completed by noticing λ in Theorem 3.7. □

Part II of proof – Dealing with variance

The proof for the variance part is two-fold. We first derive a bound on the variance, which is a function of the number of nonzero groups. We then bound the number of nonzero groups, taking into account the bound on the bias derived above.

Variance of restricted OLS: Before considering the sparse multi-site Lasso estimator, a trivial bound is shown for the variance of a restricted OLS estimation. For every subset $\psi \subset \{1, \hat{a}Ł¦, p\}$, we use it to select a subset of columns from design matrix X_i for task i. These columns form a matrix $X_{i\psi}$. Define $X_\psi = DIAG(X_{1\psi}, X_{2\psi}, ..., X_{k\psi})$, and the restricted OLS-estimator with the noise vector $\epsilon^T = (\epsilon_1, ..., \epsilon_k)^T$ is

$$\hat{\theta}^\psi = (X_\psi^T X_\psi)^{-1} X_\psi^T \epsilon \tag{A.64}$$

The ℓ_2-norm of this estimator can be bounded.

Lemma A.14. *Let m_p be a sequence with $m_p = o(\bar{n})$ and $m_p \to \infty$ for $\bar{n} \to \infty$. It holds with probability converging to 1 for $n \to \infty$*

$$\max_{\psi:|\psi|\leqslant m_p} \|\hat{\theta}^\psi\|_2^2 \leqslant \frac{2\log kp}{\bar{n}} \frac{km_p}{\phi_{\min}^2(m_p)} \sigma^2 \tag{A.65}$$

Proof. We refer the readers to Lemma 3 in Meinshausen and Yu (2009) and Lemma 3 in Liu and Zhang (2009) for the proof. Here, we again use block design structure of multi-site problem, the same as in (A.59), to obtain $\phi_{\min}(m_p)$ instead of $\phi_{\min}(km_p)$. □

The variance of the sparse multi-site Lasso estimator can be bounded by the variance of restricted OLS estimators, using bounds on the number of active groups.

Lemma A.15. *If, for a fixed value of* λ*, the number of nonzero groups of de-noised estimators* $\hat{B}^{\lambda,\xi}$ *is for every* $0 \leqslant \xi \leqslant 1$ *bounded by* m*, then*

$$\|\hat{B}^{\lambda,0} - \hat{B}^{\lambda,1}\|_F^2 \leqslant \mathcal{C} \max_{\psi:|\psi|\leqslant m} \|\hat{\theta}^{\psi}\|_2^2 \tag{A.66}$$

with $\mathcal{C}$ *as a generic constant.*

Proof. We refer the readers to Lemma 4 and Lemma 5 in Liu and Zhang (2009) for the proof. □

Let $A^p_{\lambda,\xi}$ be the set of variables in nonzero groups of the de-noised estimator $\hat{B}^{\lambda,\xi}$. Define m_p to be the largest number of nonzero groups over all values of $0 \leqslant \xi \leqslant 1$. Then we have $km_p = \sup_{0\leqslant\xi\leqslant1} |A^p_{\lambda,\xi}|$.

Lemma A.16. *Given* $0 \leqslant \alpha \leqslant 0.5$*, we have*

$$|A^p_{\lambda,\xi}|\lambda^2(1-2\alpha)^2 \leqslant \|2X^T_{A^p_{\lambda,\xi}}(Y - X\hat{\beta}^{\lambda,\xi})\|_2^2 \tag{A.67}$$

where we defined before that $X = DIAG(X_1, ..., X_k)$*,* $Y^T = (Y_1^T, ..., Y_k^T)$*.* $\hat{\beta}^{\lambda,\xi}$ *is the transpose of unfolded vector of* $\hat{B}^{\lambda,\xi}$ *by rows.* $X_{A^p_{\lambda,\xi}}$ *is* X_ψ *when* $\psi = A^p_{\lambda,\xi}$

Proof. The conditions for the solution of sparse multi-site Lasso are presented in Simon et al. (2013). We use $\hat{\beta}$ rather than $\hat{\beta}^{\lambda,\xi}$ for notational simplicity in this proof. We continue to use our notation $\hat{\beta}^j$ to refer the j-th column (here it is a group) of $\hat{B}$, and $\hat{\beta}^j_i$ to refer the i-th element (task) in $\hat{\beta}^j$. We define $X^j = DIAG(X^j_1, ..., X^j_k)$

and X_i^j to be the j-th column of X_i for task i. In other words, we allow for $(k-1)p$ number of 0 in X_i^j.

$$
\begin{aligned}
&-2{X_i^j}^T(Y - X\hat{\beta}) + \lambda\left\{\alpha\frac{\hat{\beta}_i^j}{\|\hat{\beta}_i^j\|_2} + (1-\alpha)\frac{\hat{\beta}_i^j}{\|\hat{\beta}^j\|_2/\sqrt{k}}\right\} = 0, \text{ when } \hat{\beta}_i^j \neq 0,\ \hat{\beta}^j \neq 0,\\
&-2{X_i^j}^T(Y - X\hat{\beta}) + \lambda(1-\alpha)\frac{\hat{\beta}_i^j}{\|\hat{\beta}^j\|_2/\sqrt{k}} = \lambda\alpha v_i^j, \text{ with } \|v_i^j\|_2 \leqslant 1, \text{ when } \hat{\beta}_i^j = 0,\ \hat{\beta}^j \neq 0,\\
&\left\|-2{X^j}^T(Y - X\hat{\beta})\right\|_2 \leqslant \lambda\sqrt{k}, \text{ when } \hat{\beta}^j = 0.
\end{aligned}
\tag{A.68}
$$

Let $D^P_{\lambda,\xi} = \{j \in 1, 2, ..., p | \text{group j is active for } \hat{B}^{\lambda,\xi}\}$. For each j in $D^P_{\lambda,\xi}$, we define $\hat{\beta}^j_*$ to be the vector of all $\hat{\beta}_i^j \neq 0$. Their corresponding columns X_i^js from X^j, would form a matrix X^j_*. For each j in $D^P_{\lambda,\xi}$, we define $\hat{\beta}^j_{*^C}$ to be the vector of all $\hat{\beta}_i^j = 0$. Their corresponding columns X_i^js from X^j, would form a matrix $X^j_{*^C}$. Then, from (A.68),

$$
\sum_{j=1}^{D^P_{\lambda,\xi}} \|2{X^j_*}^T(Y - X\hat{\beta})\|_2^2 \geqslant \lambda^2(1-\alpha)^2 k \sum_{j=1}^{D^P_{\lambda,\xi}} \frac{\|\hat{\beta}^j_*\|_2^2}{\|\hat{\beta}^j\|_2^2} \tag{A.69}
$$

Based on the fact that $\|a+b\|_2^2 \geqslant (\|a\|_2 - \|b\|_2)^2$

$$
\begin{aligned}
&\sum_{j=1}^{D^P_{\lambda,\xi}} \|2{X^j_{*^C}}^T(Y - X\hat{\beta})\|_2^2 \geqslant \sum_{j=1}^{D^P_{\lambda,\xi}} \left(\lambda(1-\alpha)\sqrt{k}\frac{\|\hat{\beta}^j_{*^C}\|_2}{\|\hat{\beta}^j\|_2} - \lambda\alpha\|v^j_{*^C}\|_2\right)^2\\
&= \sum_{j=1}^{D^P_{\lambda,\xi}} \left\{\lambda^2(1-\alpha)^2 k\frac{\|\hat{\beta}^j_{*^C}\|_2^2}{\|\hat{\beta}^j\|_2^2} + \lambda^2\alpha^2\|v^j_{*^C}\|_2^2 - 2\lambda^2\alpha(1-\alpha)\sqrt{k}\frac{\|\hat{\beta}^j_{*^C}\|_2}{\|\hat{\beta}^j\|_2}\|v^j_{*^C}\|_2\right\}\\
&\geqslant \sum_{j=1}^{D^P_{\lambda,\xi}} \left\{\lambda^2(1-\alpha)^2 k\frac{\|\hat{\beta}^j_{*^C}\|_2^2}{\|\hat{\beta}^j\|_2^2} + \lambda^2\alpha^2\|v^j_{*^C}\|_2^2 - \lambda^2\alpha(1-\alpha)\left[k\frac{\|\hat{\beta}^j_{*^C}\|_2^2}{\|\hat{\beta}^j\|_2^2} + \|v^j_{*^C}\|_2^2\right]\right\}\\
&= \lambda^2(1-\alpha)(1-2\alpha)k\sum_{j=1}^{D^P_{\lambda,\xi}} \frac{\|\hat{\beta}^j_{*^C}\|_2^2}{\|\hat{\beta}^j\|_2^2} - \lambda^2\alpha(1-2\alpha)\sum_{j=1}^{D^P_{\lambda,\xi}} \|v^j_{*^C}\|_2^2
\end{aligned}
\tag{A.70}
$$

Based on (A.69) and (A.70), we have

$$\|2X^T_{A^P_{\lambda,\xi}}(Y - X\hat{\beta})\|_2^2 = \sum_{j=1}^{D^P_{\lambda,\xi}} \|2X^j{}^T(Y - X\hat{\beta})\|_2^2 = \sum_{j=1}^{D^P_{\lambda,\xi}} \|2X^j_*{}^T(Y - X\hat{\beta})\|_2^2 + \sum_{j=1}^{D^P_{\lambda,\xi}} \|2X^j_{*^C}{}^T(Y - X\hat{\beta})\|_2^2 \tag{A.71}$$

$$\geqslant \lambda^2(1-\alpha)^2 k \sum_{j=1}^{D^P_{\lambda,\xi}} \frac{\|\hat{\beta}^j_*\|_2^2}{\|\hat{\beta}^j\|_2^2} + \lambda^2(1-\alpha)(1-2\alpha)k \sum_{j=1}^{D^P_{\lambda,\xi}} \frac{\|\hat{\beta}^j_{*^C}\|_2^2}{\|\hat{\beta}^j\|_2^2} - \lambda^2\alpha(1-2\alpha) \sum_{j=1}^{D^P_{\lambda,\xi}} \|v^j_{*^C}\|_2^2 \tag{A.72}$$

$$\geqslant \lambda^2(1-\alpha)(1-2\alpha)k \sum_{j=1}^{D^P_{\lambda,\xi}} \frac{\|\hat{\beta}^j_*\|_2^2 + \|\hat{\beta}^j_{*^C}\|_2^2}{\|\hat{\beta}^j\|_2^2} - \lambda^2\alpha(1-2\alpha) \sum_{j=1}^{D^P_{\lambda,\xi}} \|v^j_{*^C}\|_2^2 \tag{A.73}$$

$$\geqslant \lambda^2(1-\alpha)(1-2\alpha)k|D^P_{\lambda,\xi}| - \lambda^2\alpha(1-2\alpha)k|D^P_{\lambda,\xi}| \tag{A.74}$$

$$= \lambda^2(1-2\alpha)^2 k|D^P_{\lambda,\xi}| = \lambda^2(1-2\alpha)^2|A^P_{\lambda,\xi}| \tag{A.75}$$

□

The next lemma provides an asymptotic upper bound on the number of selected variables, the proof of which is similar to Lemma 5 in Meinshausen and Yu (2009).

Lemma A.17. *Assume conditions in Theorem 3.7, with probability converging to 1 for* $n \to \infty$,

$$\sup_{0 \leqslant \xi \leqslant 1} |A^P_{\lambda,\xi}| \leqslant \log(\bar{n}) \left\{ \left(1 + \frac{\alpha}{1-2\alpha}\right) \sqrt{k s_p} + \frac{\alpha}{1-2\alpha} \sqrt{s_h} \right\} \tag{A.76}$$

Follow from Lemmas A.14,A.15, and A.17, the next lemma bounds the variance part of the sparse multi-sites Lasso estimator:

Lemma A.18. *Assume conditions in Theorem3.7, there exists a constant* $\omega_2 > 0$, *with probability converging to 1 for* $n \to \infty$,

$$\|B^\lambda - \hat{B}^\lambda\|_F^2 \leqslant \omega_2 \sigma^2 \frac{k\bar{s}\log(kp)}{\bar{n}} \tag{A.77}$$

The lemma A.13 and A.18 together complete the proof of Theorem 3.7

Proof of Theorem 3.8:

Theorem A.19. *Let* $0.4 \leqslant \tilde{\alpha} \leqslant 1$. *Assume there exist constants* $0 \leqslant \rho_{min} \leqslant \rho_{max} \leqslant \infty$ *such that*

$$\begin{aligned} &\liminf_{n\to\infty} \phi_{min}\left(s_h \log \bar{n}\left(1+\frac{(1-\tilde{\alpha})}{\tilde{\alpha}}\right)^2\right) \geqslant \rho_{min} \\ &\limsup_{n\to\infty} \phi_{max}(s_h + \min\{\sum_{i=1}^{k} n_i, kp\}) \leqslant \rho_{max}. \end{aligned} \tag{A.78}$$

Then, for $\tilde{\lambda} \propto \sigma\sqrt{\bar{n}\log(kp)}$, *there exists* $\omega > 0$ *such that, with probability converging to 1 for* $n \to \infty$, *we have* (3.15) *with* $\tilde{s} = \{(1-\tilde{\alpha})\sqrt{s_p/k} + \tilde{\alpha}\sqrt{s_h/k}\}^2$ *instead of* $\bar{s}$.

The proof is similar to that of Theorem 3.7. Recall that in this case, however, we do not penalize $\sqrt{k}$ on group penalty. Hence, we have the following result about bias contribution of Theorem 3.8.

Lemma A.20. *Assume conditions in Theorem 3.8. The Frobenius norm of* Γ^λ *is then bounded for sufficiently large values of* $\bar{n}$, *given a constant* $\omega_1 > 0$, *by*

$$\|\Gamma^\lambda\|_F^2 \leqslant \omega_1\sigma^2\frac{k\tilde{s}\log(kp)}{\bar{n}} \tag{A.79}$$

Proof. The proof procedure is same as Lemma A.13. But instead of (A.53), we now have

$$\Lambda(\Gamma^\lambda) \leqslant 2\{(1-\tilde{\alpha})\sqrt{s_p} + \tilde{\alpha}\sqrt{s_h}\}\|\gamma^\lambda\|_2 = 2\sqrt{k\tilde{s}}\|\gamma^\lambda\|_2 \tag{A.80}$$

because we do not have $\sqrt{k}$ penalization on group penalty. Hence, in Lemma A.20, we have $\tilde{s} = \{(1-\tilde{\alpha})\sqrt{s_p/k} + \tilde{\alpha}\sqrt{s_h/k}\}^2$, instead of $\bar{s} = \{(1-\tilde{\alpha})\sqrt{s_p} + \tilde{\alpha}\sqrt{s_h/k}\}^2$. □

For restricted OLS estimation, we redefine few things here. For every subset $\psi \subset \{1, ..., kp\}$ with $|\psi| \leqslant \sum_{i=1}^{k} n_i$, we define X_ψ to be the combination of columns from design matrix X, where $X = DIAG(X_1, X_2, ..., X_k)$. The restricted OLS-estimator of

the noise vector $\epsilon^T = (\epsilon_1, ..., \epsilon_k)^T$ is then given by,

$$\hat{\theta}^{\psi} = (X_{\psi}^T X_{\psi})^{-1} X_{\psi}^T \epsilon \tag{A.81}$$

For the variance contribution, the proof is similar to that of Theorem 3.7. We present the required Lemmas for Theorem 3.8 here.

Lemma A.21. *Let m_n be a sequence with $m_n = o(k\bar{n})$ and $m_n \to \infty$ for $\bar{n} \to \infty$. It holds with probability converging to 1 for $n \to \infty$*

$$\max_{\psi:|\psi|\leqslant m_n} \|\hat{\theta}^{\psi}\|_2^2 \leqslant \frac{2\log kp}{\bar{n}} \frac{m_n}{\phi_{\min}^2(m_n)} \sigma^2 \tag{A.82}$$

Lemma A.22. *If, for a fixed value of λ, the number of active variables of de-noised estimators $\hat{B}^{\lambda,\xi}$ is for every $0 \leqslant \xi \leqslant 1$ bounded by m, then*

$$\|\hat{B}^{\lambda,0} - \hat{B}^{\lambda,1}\|_F^2 \leqslant \mathcal{C} \max_{\psi:|\psi|\leqslant m} \|\hat{\theta}^{\psi}\|_2^2 \tag{A.83}$$

with $\mathcal{C}$ as a generic constant.

Let $A_{\lambda,\xi}^1$ be the set of active variables of the de-noised estimator $\hat{B}^{\lambda,\xi}$. Let m_n to be the largest number of active variables over all values of $0 \leqslant \xi \leqslant 1$. Then we have $m_n = \sup_{0\leqslant\xi\leqslant 1} |A_{\lambda,\xi}^1|$.

Lemma A.23. *For any $0 \leqslant \alpha \leqslant 1$, we have*

$$|A_{\lambda,\xi}^1| \lambda^2 \alpha^2 \leqslant \|2X_{A_{\lambda,\xi}^1}^T (Y - X\hat{\beta}^{\lambda,\xi})\|_2^2 \tag{A.84}$$

where we defined before that $X = DIAG(X_1, ..., X_k)$, $Y^T = (Y_1^T, ..., Y_k^T)$. $\hat{\beta}^{\lambda,\xi}$ is the transpose of unfolded vector of $\hat{B}^{\lambda,\xi}$ by rows. $X_{A_{\lambda,\xi}^1}$ is X_{ψ} when $\psi = A_{\lambda,\xi}^1$

Lemma A.24. *Assume conditions in Theorem 3.8, with probability converging to 1 for $n \to \infty$,*

$$\sup_{0\leqslant\xi\leqslant 1} |A_{\lambda,\xi}^1| \leqslant \log(\bar{n}) \left\{ \sqrt{s_h} + \frac{1-\alpha}{\alpha} \sqrt{s_p} \right\} \tag{A.85}$$

Lemma A.25. *Assume conditions in Theorem3.8, there exists a constant* $\omega_2 > 0$*, with probability converging to 1 for* $n \to \infty$*,*

$$\|B^{\lambda} - \hat{B}^{\lambda}\|_F^2 \leqslant \omega_2 \sigma^2 \frac{k\tilde{s}\log(kp)}{\bar{n}} \tag{A.86}$$

Lemma A.20 and Lemma A.25 complete the proof of Theorem 3.8

A.3 Proofs for Theorems in Chapter 4

In this section, we give the proofs for Theorem 4.4, the two examples in Subsection 4.4, Theorems 4.9 and Theorem 4.10 (which are the key results used in the proof of Theorem 4.3). Then proofs for other Lemmas are presented in Subsection A.3.

Proof of Theorem 4.4

The outline of this proof is the same as the outline of the proof for Theorem 4.3. The key difference here is that, given a ϕ-mixing process with $0.781 \leqslant r_\phi \leqslant 2$, we are able to derive sharper rates for Theorem 4.10 and Lemma 4.13, which result in $m = T^{\frac{r_\phi}{r_\phi+2}}$. For $r_\phi \leqslant 2$ this rate is sharper since $T^{\frac{r_\phi}{r_\phi+2}} \geqslant T^{\frac{r_\phi-1}{r_\phi}}$. Specifically, using the concentration inequality from Kontorovich et al. (2008), we show two Lemmas which give us a larger m than Theorem 4.10 and Lemma 4.13.

Lemma A.26. *Define the event*

$$\mathcal{B}_{m,T} = \left\{ \sup_{j,k} \sup_{f_{j,k} \in B_H(1), \|f_{j,k}\|_2 \leqslant \gamma_m} |\|f_{j,k}\|_T - \|f_{j,k}\|_2| \leqslant \frac{\gamma_m}{2} \right\}.$$

For a stationary ϕ-mixing process $(X_t)_{t=0}^T$ with $0.781 \leqslant r_\phi \leqslant 2$ and $m = T^{\frac{r_\phi}{r_\phi+2}}$, we have $\mathbb{P}(\mathcal{B}_{m,T}) \geqslant 1 - c_2 exp(-c_3(m\gamma_m^2)^2)$ where c_2 and c_3 are constants.

Moreover, on the event $\mathcal{B}_{m,T}$, for any $g \in B_{\mathcal{H}}(1)$ with $\|g\|_2 \geqslant \gamma_m$,

$$\frac{\|g\|_2}{2} \leqslant \|g\|_T \leqslant \frac{3}{2}\|g\|_2. \tag{A.87}$$

Lemma A.27. *Given properties of γ_m and $\delta_{m,j}^2 = c_4\{\frac{s_j \log(d)}{m} + s_j\epsilon_m^2\}$, we define the event*

$$\mathcal{D}_{m,T} = \{\forall j \in [1, 2, ..., d], \|g_j\|_T \geqslant \|g_j\|_2/2, \\ \text{for all } g_j \in 2\mathcal{F}_j \text{ with } \|g_j\|_2 \geqslant \delta_{m,j}\}.$$

For a ϕ-mixing process $(X_t)_{t=0}^T$ with $0.781 \leqslant r_\phi \geqslant 2$ and $m = T^{\frac{r_\phi}{r_\phi+2}}$, we have $\mathbb{P}(\mathcal{D}_{m,T}) \geqslant 1 - c_2 exp(-c_3(m(\min_j \delta_{m,j}^2))^2)$ where c_2, c_3 and c_4 are constants.

Following the outline of the proof for Theorem 4.3, we replace Theorem 4.10 and Lemma 4.13 by Lemma A.26 and Lemma A.27, which allows us to prove Theorem 4.4.

Proofs for Subsection 4.4

Now we give proofs for Lemmas 4.5 and Lemma 4.7 for the two examples in Subsection 4.4.

Proof for Lemma 4.5. : Recall our definition of $\tilde{\varepsilon}_m$, by choosing M_0 as $M_0 = \xi$, we have that

$$\begin{aligned} &\log(dT)\left(3\frac{\log(M_0 dT)}{\sqrt{m}}\sqrt{\sum_{i=1}^{M_0}\min(\mu_i,\sigma^2)}\right. \\ &\qquad\left.+\sqrt{\frac{T}{m}}\sqrt{\sum_{i=M_0+1}^{\infty}\min(\mu_i,\sigma^2)}\right) \\ &= 3\frac{\log(\xi dT)\log(dT)}{\sqrt{m}}\sqrt{\sum_{i=1}^{\xi}\min(\mu_i,\sigma^2)}. \end{aligned} \tag{A.88}$$

Since $\min(\mu_i, \sigma^2) \leqslant \sigma^2$, that equation is upper bounded by

$$3\sigma\sqrt{\frac{\xi}{m}}\log(\xi dT)\log(dT). \tag{A.89}$$

Since $\tilde{\varepsilon}_m$ is the minimal value of σ such that (A.88) lower than σ^2, from the upper bound (A.89) we can show that

$$\tilde{\varepsilon}_m = O\left(\sqrt{\frac{\xi}{m}}\log(\xi dT)^2\right).$$

□

Proof of Lemma 4.7: Before proving $\tilde{\epsilon}_m$, we recall the discussion of ϵ_m Raskutti et al. (2012). To simplify the discussion, we assume that there exists an integer ℓ_0 such that $\sigma^2 = \frac{1}{\ell_0^{2\alpha}}$. That assumption doesn't affect the rate which we'll get for ϵ_m. Using the definition of ℓ_0, $\min(\mu_i, \sigma^2) = \sigma^2$ when $i < \ell_0$ and $\min(\mu_i, \sigma^2) = \mu_i$ when $i \geqslant \ell_0$. Therefore, since $\mu_i = (1/i)^{2\alpha}$, we have

$$\begin{aligned}
&\frac{1}{\sqrt{m}} \sqrt{\sum_{i=1}^{\infty} \min(\mu_i, \sigma^2)} \\
&\leqslant \frac{1}{\sqrt{m}} \sqrt{\ell_0 \sigma^2 + \frac{1}{2\alpha - 1} \frac{1}{\ell_0^{2\alpha-1}}} \\
&= \frac{1}{\sqrt{m}} \sigma^{1-\frac{1}{2\alpha}} \sqrt{\frac{2\alpha}{2\alpha - 1}}.
\end{aligned}$$

Hence $\epsilon_m = O(m^{-\frac{\alpha}{2\alpha+1}})$. For $\tilde{\epsilon}_m$, we still define ℓ_0 to be $\sigma^2 = \frac{1}{\ell_0^{2\alpha}}$. We require the nuisance parameter $M_0 \geqslant \ell_0$, whose value will be assigned later. Again, using the fact that $\min(\mu_i, \sigma^2) = \sigma^2$ when $i < \ell_0$ and $\min(\mu_i, \sigma^2) = \mu_i$ when $i \geqslant \ell_0$ and

$\mu_i = (1/i)^{2\alpha}$, we have

$$
\begin{aligned}
&\log(dT)\left(\sqrt{\frac{T}{m}}\sqrt{\sum_{i=M_0+1}^{\infty}\min(\mu_i,\sigma^2)}\right.\\
&\quad\left.+3\frac{\log(M_0 dT)}{\sqrt{m}}\sqrt{\sum_{i=1}^{M_0}\min(\mu_i,\sigma^2)}\right)\\
&\leqslant \log(dT)\left(\sqrt{\frac{T}{m}}\sqrt{\frac{1}{2\alpha-1}\frac{1}{M_0^{2\alpha-1}}}\right.\\
&\quad\left.+3\frac{\log(M_0 dT)}{\sqrt{m}}\sqrt{\sigma^{2-\frac{1}{\alpha}}+\frac{1}{2\alpha-1}\left(\frac{1}{\ell_0^{2\alpha-1}}-\frac{1}{M_0^{2\alpha-1}}\right)}\right)\\
&= \log(dT)\left(\sqrt{\frac{T}{m}}\sqrt{\frac{1}{2\alpha-1}\frac{1}{M_0^{2\alpha-1}}}\right.\\
&\quad\left.+3\frac{\log(M_0 dT)}{\sqrt{m}}\sqrt{\frac{2\alpha}{2\alpha-1}\sigma^{2-\frac{1}{\alpha}}-\frac{1}{2\alpha-1}\frac{1}{M_0^{2\alpha-1}}}\right)\\
&\leqslant \log(dT)\left(\sqrt{\frac{T}{m}}\sqrt{\frac{1}{2\alpha-1}}\frac{1}{M_0^{\alpha-\frac{1}{2}}}\right.\\
&\quad\left.+3\frac{\log(M_0 dT)}{\sqrt{m}}\sqrt{\frac{2\alpha}{2\alpha-1}}\sigma^{1-\frac{1}{2\alpha}}\right).
\end{aligned}
$$

In order to obtain a similar rate as ϵ_m, we set up the value of M_0 such that $\left(\sqrt{\sqrt{\frac{T}{m}}\frac{1}{M_0^{\alpha-\frac{1}{2}}}}\right)^{1+\frac{1}{2\alpha}} = \frac{1}{\sqrt{m}}$. In other words, $M_0 = m^{\frac{1}{2\alpha+1}}T^{\frac{1}{2\alpha-1}}$. After plugging in the value of M_0, we obtain an upper bound

$$
\begin{aligned}
&\log(dT)\left(3\sqrt{\frac{2\alpha}{2\alpha-1}}\frac{\log(M_0 dT)}{\sqrt{m}}\sigma^{1-\frac{1}{2\alpha}}\right.\\
&\qquad\left.+\sqrt{\frac{1}{2\alpha-1}}\left(\frac{1}{\sqrt{m}}\right)^{\frac{4\alpha}{2\alpha+1}}\right).
\end{aligned}
\tag{A.90}
$$

Compare the upper bound (A.90) with σ^2, we obtain

$$\begin{aligned}\tilde{\epsilon}_m &= O(m^{-\frac{\alpha}{2\alpha+1}}(\log(M_0 dT)\log(dT))^{\frac{2\alpha}{2\alpha+1}}) \\ &= O\left(\left(\frac{\log(dT)^2}{\sqrt{m}}\right)^{\frac{2\alpha}{2\alpha+1}}\right).\end{aligned}$$

□

Proof of Theorem 4.9

We consider single univariate function here and use f to refer each $f_{j,k}$. Finally, we'll use union bound to show that the result holds for every j, k. Before presenting the proof, we point out that there exists an equivalent class $\mathbb{F}$, which means that

$$\begin{aligned}\sup_{\|f\|_{\mathcal{H}}\leqslant 1, \|f\|_2\leqslant\sigma}\left|\frac{1}{T}\sum_{t=1}^{T} f(X_t)w_t\right| &\geqslant \sup_{f\in\mathbb{F}}\left|\frac{1}{T}\sum_{t=1}^{T} f(X_t)w_t\right|, \\ \sup_{\|f\|_{\mathcal{H}}\leqslant 1, \|f\|_2\leqslant\sigma}\left|\frac{1}{T}\sum_{t=1}^{T} f(X_t)w_t\right| &\leqslant \sup_{f\in\sqrt{2}\mathbb{F}}\left|\frac{1}{T}\sum_{t=1}^{T} f(X_t)w_t\right|.\end{aligned} \tag{A.91}$$

That function class $\mathbb{F}$ is defined as

$$\mathbb{F} = \{f = \sum_{i=1}^{\infty}\beta_i\sqrt{\mu_i}\Phi_i(x) | \sum_{i=1}^{\infty}\eta_i\beta_i^2 \leqslant 1\},$$

where $\eta_i = \left(\min\left(1, \frac{\sigma^2}{\mu_i}\right)\right)^{-1}$. The equivalence is because of

$$\begin{aligned}&\{f | \|f\|_{\mathcal{H}} \leqslant 1, \|f\|_2 \leqslant \sigma\} \\ &= \{f = \sum_{i=1}^{\infty}\beta_i\sqrt{\mu_i}\Phi_i(x) | \sum_{i=1}^{\infty}\beta_i^2 \leqslant 1, \sum_{i=1}^{\infty}\mu_i\beta_i^2 \leqslant \sigma^2\},\end{aligned}$$

(1) $\sum_{i=1}^{\infty}\max(1, \frac{\mu_i}{\sigma^2})\beta_i^2 \leqslant 1 \Rightarrow \sum_{i=1}^{\infty}\beta_i^2 \leqslant 1, \sum_{i=1}^{\infty}\frac{\mu_i}{\sigma^2}\beta_i^2 \leqslant 1$, and (2) $\sum_{i=1}^{\infty}\beta_i^2 \leqslant 1$, $\sum_{i=1}^{\infty}\frac{\mu_i}{\sigma^2}\beta_i^2 \leqslant 1 \Rightarrow \sum_{i=1}^{\infty}\max(1, \frac{\mu_i}{\sigma^2})\beta_i^2 \leqslant \sum_{i=1}^{\infty}(1 + \frac{\mu_i}{\sigma^2})\beta_i^2 \leqslant 2$. Next, we prove the

results for $f \in \mathbb{F}$. Let's define

$$Y_n = \frac{1}{T}\sum_{t=1}^{n} \Phi_i(X_t) w_t. \tag{A.92}$$

Then we have

$$\begin{aligned} Y_n - Y_{n-1} =& \frac{1}{T}\Phi_i(X_n) w_n, \\ E[Y_n - Y_{n-1} | w_1, ..., w_{n-1}] =& \frac{1}{T} E[\Phi_i(X_n) | w_1, ..., w_{n-1}] E[w_n] = 0. \end{aligned}$$

It tells us that $\{Y_n\}_{n=1}^{T}$ is a martingale. Therefore, we are able to use Lemma 4 on Page 20 in Hall et al. (2016). Additionally, given that $\Phi_i(.)$ is bounded by 1 and Assumption 1 for w_t, we know that

$$|Y_n - Y_{n-1}| = \frac{1}{T}|\Phi_i(X_n) w_n| \leqslant \frac{\log(dT)}{T}.$$

In order to use Lemma 4 in Hall et al. (2016), we bound the so-called term M_n^i and hence the so-called summation term D_n in Hall et al. (2016), which are

$$M_n^i = \sum_{t=1}^{n} E\left[(Y_t - Y_{t-1})^i | w_1, ..., w_{t-1}\right] \leqslant n\frac{\log(dT)^i}{T^i},$$

and

$$\begin{aligned} D_n &= \sum_{i \geqslant 2} \frac{\rho^i}{i!} M_n^i \leqslant \sum_{i \geqslant 2} \frac{\rho^i}{i!} n \frac{\log(dT)^i}{T^i} \\ &= n\left(e^{\rho \log(dT)/T} - 1 - \frac{\rho \log(dT)}{T}\right), \end{aligned}$$

for any nuisance parameter ρ. That bound on D_n is defined as $\hat{D}_n$. Then using the results from Lemma 4 in Hall et al. (2016) that $\max(E[e^{\rho Y_n}], E[e^{-\rho Y_n}]) \leqslant e^{\hat{D}_n}$ for a martingale $\{Y_n\}_{n=1}^{T}$ and the Markov inequality, we are able to get an upper bound

on the desired quantity Y_n, that is,

$$\begin{aligned} &P(|Y_n| \geqslant y) \\ &\leqslant E[e^{\rho|Y_n|}]e^{-\rho y} \leqslant \left(E[e^{\rho Y_n}] + E[e^{-\rho Y_n}]\right) e^{-\rho y} \\ &\leqslant 2e^{\hat{D}_n - \rho y} \\ &= 2\exp(n(e^{\rho \log(dT)/T} - 1 - \frac{\rho \log(dT)}{T}) - \rho y). \end{aligned} \tag{A.93}$$

By setting the nuisance parameter $\rho = \frac{T}{\log(dT)} \log(\frac{yT}{n \log(dT)} + 1)$, that yields the lowest bound

$$P(|Y_n| \geqslant y) \leqslant 2\exp\left(-nH\left(\frac{Ty}{n \log(dT)}\right)\right),$$

where $H(x) = (1+x)\log(1+x) - x$. We can use the fact that $H(x) \geqslant \frac{3x^2}{2(x+3)}$ for $x \geqslant 0$ to further simplify the bound and get

$$P(|Y_n| \geqslant y) \leqslant 2\exp\left(\frac{-3T^2y^2}{2Ty \log(dT) + 6n \log(dT)^2}\right).$$

Plugging in the definition of Y_n, this result means that

$$\begin{aligned} &P\left(\left|\frac{1}{T}\sum_{t=1}^{n} \Phi_i(X_t) w_t\right| \geqslant y\right) \\ &\leqslant 2\exp\left(\frac{-3T^2y^2}{2Ty \log(dT) + 6n \log(dT)^2}\right). \end{aligned}$$

Then by setting $n = T$, we get

$$\begin{aligned} &P\left(\left|\frac{1}{T}\sum_{t=1}^{T} \Phi_i(X_t) w_t\right| \geqslant y\right) \\ &\leqslant 2\exp\left(\frac{-3Ty^2}{2y \log(dT) + 6 \log(dT)^2}\right). \end{aligned} \tag{A.94}$$

Using union bound, we obtain an upper bound for the supreme over M_0 such terms, which is

$$\begin{aligned} &P\left(\sup_{i=1,2,..,M_0}\left|\frac{1}{T}\sum_{t=1}^{T}\Phi_i(X_t)w_t\right| \geqslant y\right) \\ &\leqslant \exp\left(\frac{-3Ty^2}{2y\log(dT)+6\log(dT)^2}+\log(2M_0)\right). \end{aligned} \tag{A.95}$$

We will show next that (A.95) enables us to bound $\sup_{f\in F}\left|\frac{1}{T}\sum_{t=1}^{T} f(X_t)w_t\right|$, which is our goal. First, we decompose it into two parts

$$\begin{aligned} &\sup_{f\in\mathbb{F}}\left|\frac{1}{T}\sum_{t=1}^{T} f(X_t)w_t\right| \\ &= \sup_{\sum_{i=1}^{\infty}\eta_i\beta_i^2\leqslant 1}\left|\frac{1}{T}\sum_{t=1}^{T}\left(\sum_{i=1}^{\infty}\beta_i\sqrt{\mu_i}\Phi_i(X_t)\right)w_t\right| \\ &= \sup_{\sum_{i=1}^{\infty}\eta_i\beta_i^2\leqslant 1}\left|\frac{1}{T}\sum_{i=1}^{\infty}\beta_i\sqrt{\mu_i}\left(\sum_{t=1}^{T}\Phi_i(X_t)w_t\right)\right| \\ &\leqslant \sup_{\sum_{i=1}^{\infty}\eta_i\beta_i^2\leqslant 1}\left|\frac{1}{T}\sum_{i=1}^{M_0}\beta_i\sqrt{\mu_i}\left(\sum_{t=1}^{T}\Phi_i(X_t)w_t\right)\right| \\ &+ \sup_{\sum_{i=1}^{\infty}\eta_i\beta_i^2\leqslant 1}\left|\frac{1}{T}\sum_{i=M_0+1}^{\infty}\beta_i\sqrt{\mu_i}\left(\sum_{t=1}^{T}\Phi_i(X_t)w_t\right)\right|. \end{aligned}$$

The second part can be easily bounded using Assumption 1 in following

$$\begin{aligned} &\sup_{\sum_{i=1}^{\infty}\eta_i\beta_i^2\leqslant 1}\left|\frac{1}{T}\sum_{i=M_0+1}^{\infty}\beta_i\sqrt{\mu_i}\left(\sum_{t=1}^{T}\Phi_i(X_t)w_t\right)\right| \\ &\leqslant \sup_{\sum_{i=1}^{\infty}\eta_i\beta_i^2\leqslant 1}\sum_{i=M_0+1}^{\infty}\beta_i\sqrt{\mu_i}\log(dT). \end{aligned}$$

Using Cauchy-Schwarz inequality, this upper bound is further bounded by

$$\sup_{\sum_{i=1}^{\infty}\eta_i\beta_i^2\leqslant 1}\sqrt{\sum_{i=M_0+1}^{\infty}\eta_i\beta_i^2}\sqrt{\sum_{i=M_0+1}^{\infty}\frac{\mu_i}{\eta_i}}\log(dT),$$

which is smaller than

$$\sqrt{\sum_{i=M_0+1}^{\infty}\frac{\mu_i}{\eta_i}}\log(dT)=\sqrt{\sum_{i=M_0+1}^{\infty}\min(\mu_i,\sigma^2)}\log(dT).$$

Our next goal is to show that we can bound the first part

$$\sup_{\sum_{i=1}^{\infty}\eta_i\beta_i^2\leqslant 1}\left|\frac{1}{T}\sum_{i=1}^{M_0}\beta_i\sqrt{\mu_i}\left(\sum_{t=1}^{T}\Phi_i(X_t)w_t\right)\right|$$

using (A.95). To bound that, simply using Cauchy-Schwarz inequality, we get

$$\begin{aligned}
&\sup_{\sum_{i=1}^{\infty}\eta_i\beta_i^2\leqslant 1}\left|\frac{1}{T}\sum_{i=1}^{M_0}\beta_i\sqrt{\mu_i}\left(\sum_{t=1}^{T}\Phi_i(X_t)w_t\right)\right|\\
&\leqslant\sup_{\sum_{i=1}^{\infty}\eta_i\beta_i^2\leqslant 1}\sqrt{\sum_{i=1}^{M_0}\eta_i\beta_i^2}\sqrt{\sum_{i=1}^{M_0}\frac{\mu_i}{\eta_i}\left(\frac{1}{T}\sum_{t=1}^{T}\Phi_i(X_t)w_t\right)^2}\\
&\leqslant\sqrt{\sum_{i=1}^{M_0}\frac{\mu_i}{\eta_i}}\sup_{i=1,2,..,M_0}\left|\frac{1}{T}\sum_{t=1}^{T}\Phi_i(X_t)w_t\right|.
\end{aligned}$$

Using (A.95), we show that the first part is upper bounded by

$$\sqrt{\sum_{i=1}^{M_0}\min(\mu_i,\sigma^2)}y,$$

with probability at least $1 - \exp\left(\frac{-3Ty^2}{2y\log(dT)+6\log(dT)^2} + \log(M_0)\right)$. Therefore, after combining the bounds on the two parts, we obtain the upper bound for

$$\sup_{f\in\sqrt{2}\mathbb{F}} \left| \frac{1}{T}\sum_{t=1}^{T} f(X_t)w_t \right|$$

, which is

$$\sup_{f\in\sqrt{2}\mathbb{F}} \left| \frac{1}{T}\sum_{t=1}^{T} f(X_t)w_t \right| \leqslant \sqrt{2}\left(\sqrt{\sum_{i=1}^{M_0} \min(\mu_i, \sigma^2)y} + \sqrt{\sum_{i=M_0+1}^{\infty} \min(\mu_i, \sigma^2)\log(dT)} \right),$$

with probability at least $1 - \exp\left(\frac{-3Ty^2}{2y\log(dT)+6\log(dT)^2} + \log(M_0)\right)$. Further, after applying union bound on all $(j,k) \in \{1,2,...,d\}^2$ and recalling the connection between $\{f | \|f\|_{\mathcal{H}} \leqslant 1, \|f\|_2 \leqslant \sigma\}$ and $\sqrt{2}\mathbb{F}$, we can show that with probability at least $1 - \exp\left(\frac{-3Ty^2}{2y\log(dT)+6\log(dT)^2} + \log(M_0) + 2\log(d)\right)$,

$$\sup_{\|f_{j,k}\|_{\mathcal{H}}\leqslant 1 \|f_{j,k}\|_2 \leqslant \sigma} \left| \frac{1}{T}\sum_{t=1}^{T} f_{j,k}(X_t)w_{t,j} \right| \leqslant \sqrt{2}\left(\sqrt{\sum_{i=1}^{M_0} \min(\mu_i, \sigma^2)y} + \sqrt{\sum_{i=M_0+1}^{\infty} \min(\mu_i, \sigma^2)\log(dT)} \right),$$

for all $(j,k) \in \{1,2,...,d\}^2$ and any M_0, y.

Finally, by setting $y = 3\frac{(\log(M_0 dT))\log(dT)}{\sqrt{T}}$, we obtain that, with probability at

least $1 - \frac{1}{M_0 T}$,

$$\begin{aligned}
&\sup_{\|f_{j,k}\|_{\mathcal{H}} \leqslant 1, \|f_{j,k}\|_2 \leqslant \sigma} \left| \frac{1}{T} \sum_{t=1}^{T} f_{j,k}(X_t) w_t \right| \\
&\leqslant \sqrt{2} \log(dT) \left(3 \frac{(\log(M_0 dT))}{\sqrt{T}} \sqrt{\sum_{i=1}^{M_0} \min(\mu_i, \sigma^2)} \right. \\
&\qquad \left. + \sqrt{\sum_{i=M_0+1}^{\infty} \min(\mu_i, \sigma^2)} \right).
\end{aligned}$$

Here, we assumed $T \geqslant 2$, $\log(M_0 dT) \geqslant 1$. Our definition of $\tilde{\epsilon}_m$ guarantees that, if $\sigma > \tilde{\epsilon}_m$, then

$$\begin{aligned}
&\sqrt{2} \log(dT) \left(3 \frac{(\log(M_0 dT))}{\sqrt{T}} \sqrt{\sum_{i=1}^{M_0} \min(\mu_i, \sigma^2)} \right. \\
&\qquad \left. + \sqrt{\sum_{i=M_0+1}^{\infty} \min(\mu_i, \sigma^2)} \right) \\
&\leqslant \sqrt{2} \sqrt{\frac{m}{T}} \sigma^2.
\end{aligned}$$

That completes our proof for Theorem 4.9.

Proof of Theorem 4.10

Since we have $\|f_{j,k}\|_\infty \leqslant 1$, it suffices to bound

$$\mathbb{P}\left(\sup_{j,k} \sup_{f_{j,k} \in B_{\mathcal{H}}(1), \|f_{j,k}\|_2 \leqslant \gamma_m} |\|f_{j,k}\|_T^2 - \|f_{j,k}\|_2^2| \geqslant \frac{\gamma_m^2}{4} \right).$$

The proofs are based on the result for independent case from Lemma 7 in Raskutti et al. (2012), which shows that there exists constants $(\tilde{c}_1, \tilde{c}_2)$ such that

$$\begin{aligned}&\mathbb{P}_0\left(\sup_{j,k}\sup_{f_{j,k}\in B_H(1),\|f_{j,k}\|_2\leqslant\gamma_m}|\frac{1}{m}\sum_{t=1}^{m}f_{j,k}^2(\tilde{X}_t)-\|f_{j,k}\|_2^2|\geqslant\frac{\gamma_m^2}{10}\right)\\&\leqslant\tilde{c}_1\exp(-\tilde{c}_2 m\gamma_m^2),\end{aligned}\tag{A.96}$$

where $\{\tilde{X}_t\}_{t=1}^m$ are i.i.d drawn from the stationary distribution of X_t denoted by $\mathbb{P}_0$. Let $T = m\ell$. We divide the stationary T-sequence $X_T = (X_1, X_2, ..., X_T)$ into m blocks of length ℓ. We use $X_{a,b}$ to refer the b-th variable in block a. Therefore, we can rewrite

$$\mathbb{P}\left(\sup_{j,k}\sup_{f_{j,k}\in B_{\mathcal{H}}(1),\|f_{j,k}\|_2\leqslant\gamma_m}|\frac{1}{T}\sum_{t=1}^{T}f_{j,k}^2(X_t)-\|f_{j,k}\|_2^2|\geqslant\frac{\gamma_m^2}{4}\right)$$

as

$$\begin{aligned}&\mathbb{P}\Bigg(\sup_{j,k}\sup_{f_{j,k}\in B_{\mathcal{H}}(1),\|f_{j,k}\|_2\leqslant\gamma_m}\\&\quad|\frac{1}{\ell}\sum_{b=1}^{\ell}\frac{1}{m}\sum_{a=1}^{m}f_{j,k}^2(X_{a,b})-\|f_{j,k}\|_2^2|\geqslant\frac{\gamma_m^2}{4}\Bigg).\end{aligned}\tag{A.97}$$

Using the fact that $\sup|\sum ..| \leqslant \sum \sup|..|$, (A.97) is smaller than

$$\begin{aligned}&\mathbb{P}\Bigg(\frac{1}{\ell}\sum_{b=1}^{\ell}\sup_{j,k}\sup_{f_{j,k}\in B_{\mathcal{H}}(1),\|f_{j,k}\|_2\leqslant\gamma_m}\\&\quad|\frac{1}{m}\sum_{a=1}^{m}f_{j,k}^2(X_{a,b})-\|f_{j,k}\|_2^2|\geqslant\frac{\gamma_m^2}{4}\Bigg),\end{aligned}$$

which, by using the fact that $P(\frac{1}{\ell}\sum_{i=1}^{\ell} a_i \geqslant c) \leqslant P(\cup_{i=1}^{\ell}(a_i \geqslant c)) \leqslant \sum_{i=1}^{\ell} P(a_i \geqslant c)$, is bounded by

$$\sum_{b=1}^{\ell} \mathbb{P}\Bigg(\sup_{j,k} \sup_{f_{j,k}\in B_{\mathcal{H}}(1), \|f_{j,k}\|_2 \leqslant \gamma_m} |\frac{1}{m}\sum_{a=1}^{m} f_{j,k}^2(X_{a,b}) - \|f_{j,k}\|_2^2| \geqslant \frac{\gamma_m^2}{4} \Bigg).$$

Using the fact that the process is stationary, it is equivalent to

$$\ell\mathbb{P}\Bigg(\sup_{j,k} \sup_{f_{j,k}\in B_{H}(1), \|f_{j,k}\|_2 \leqslant \gamma_m} |\frac{1}{m}\sum_{a=1}^{m} f_{j,k}^2(X_{a,\ell}) - \|f_{j,k}\|_2^2| \geqslant \frac{\gamma_m^2}{4} \Bigg). \tag{A.98}$$

Our next steps are trying to bound the non-trivial part in (A.98). Because of Lemma 2 in Nobel and Dembo (1993), we can replace $\{X_{a,l}\}_{a=1}^{m}$ by their independent copies under probability measure P_0 with a sacrifice of $m\beta(\ell)$. Then we are able to use (A.96) to bound the remaining probability. First, using Lemma 2 in Nobel and Dembo (1993), we have

$$\begin{aligned}
&\mathbb{P}\Bigg(\sup_{j,k} \sup_{f_{j,k}\in B_{H}(1), \|f_{j,k}\|_2 \leqslant \gamma_m} |\frac{1}{m}\sum_{a=1}^{m} f_{j,k}^2(X_{a,\ell}) - \|f_{j,k}\|_2^2| \geqslant \frac{\gamma_m^2}{4} \Bigg) \\
&\leqslant \mathbb{P}_0\Bigg(\sup_{j,k} \sup_{f_{j,k}\in B_{\mathcal{H}}(1), \|f_{j,k}\|_2 \leqslant \gamma_m} |\frac{1}{m}\sum_{a=1}^{m} f_{j,k}^2(X_{a,\ell}) - \|f_{j,k}\|_2^2| \geqslant \frac{\gamma_m^2}{4} \Bigg) + m\beta(\ell).
\end{aligned}$$

Now, using (A.96), it is bounded by

$$\tilde{c}_1 \exp(2\log(d) - \tilde{c}_2 m\gamma_m^2) + m\beta(\ell).$$

Therefore, we get

$$\begin{aligned}
&\mathbb{P}\Bigg(\sup_{j,k} \sup_{f_{j,k}\in B_{\mathcal{H}}(1), \|f_{j,k}\|_2 \leqslant \gamma_m} \\
&\qquad |\frac{1}{T}\sum_{t=1}^{T} f_{j,k}^2(X_t) - \|f_{j,k}\|_2^2| \geqslant \frac{\gamma_m^2}{4} \Bigg) \\
&\leqslant \ell \mathbb{P}\Bigg(\sup_{j,k} \sup_{f_{j,k}\in B_{\mathcal{H}}(1), \|f_{j,k}\|_2 \leqslant \gamma_m} \\
&\qquad |\frac{1}{m}\sum_{a=1}^{m} f_{j,k}^2(X_{a,\ell}) - \|f_{j,k}\|_2^2| \geqslant \frac{\gamma_m^2}{4} \Bigg) \\
&\leqslant \ell \tilde{c}_1 \exp(2\log(d) - \tilde{c}_2 m\gamma_m^2) + T\beta(\ell).
\end{aligned}$$

Recall that $\ell = T/m$ and the definition of $\beta(\ell)$, which is equal to ℓ^{-r_β}, that bound hence is

$$\tilde{c}_1 \exp(2\log(dT) - \tilde{c}_2 m\gamma_m^2) + T\left(\frac{T}{m}\right)^{-r_\beta}.$$

Recall our definition of γ_m with $m\gamma_m^2 \geqslant c_1^2 \log(dT)$ and $m = T^{\frac{c_0 r_\beta - 1}{c_0 r_\beta}}$, hence that probability is

$$\left(\tilde{c}_3 \exp(-\tilde{c}_4 m\gamma_m^2) + T^{-\left(\frac{1-c_0}{c_0}\right)} \right),$$

for some constants $\tilde{c}_3$ and $\tilde{c}_4$. That completes the proof. For the follow-up statement, condition on the event $\mathcal{B}_{m,T}$, for any $g \in B_{\mathcal{H}}(1)$ with $\|g\|_2 \geqslant \gamma_m$, we have $h = \gamma_m \frac{g}{\|g\|_2}$ is in $B_{\mathcal{H}}(1)$ and $\|h\|_2 \leqslant \gamma_m$. Therefore, we have

$$\left| \|\gamma_m \frac{g}{\|g\|_2}\|_T - \|\gamma_m \frac{g}{\|g\|_2}\|_2 \right| \leqslant \frac{\gamma_m}{2},$$

which implies

$$|\|g\|_T - \|g\|_2| \leqslant \frac{1}{2}\|g\|_2.$$

Other proofs

Proof of Lemma 4.11. The statement which we want to show is equivalent to

$$|\frac{1}{T}\sum_{t=1}^{T} f_{j,k}(X_t)w_{t,j}| \leqslant 4\sqrt{2}\|f_{j,k}\|_{\mathcal{H}}\sqrt{\frac{m}{T}}(\tilde{\gamma}_m^2 + \tilde{\gamma}_m\frac{\|f_{j,k}\|_T}{\|f_{j,k}\|_{\mathcal{H}}}) \tag{A.99}$$

for any $f_{j,k} \in \mathcal{H}$, for any $(j,k) \in [1,2,...,d]^2$.
For each j,k, we define

$$Z_{T,j,k}(w;\ell) := \left| \sup_{\|f_{j,k}\|_T \leqslant \ell, \|f_{j,k}\|_{\mathcal{H}} \leqslant 1} \frac{1}{T}\sum_{t=1}^{T} f_{j,k}(X_t)w_{t,j} \right|.$$

We claim that on event $\mathcal{A}_{m,T} \cap \mathcal{B}_{m,T}$,

$$Z_{T,j,k}(w;\ell) \leqslant 4\sqrt{2}\sqrt{\frac{m}{T}}(\tilde{\gamma}_m^2 + \tilde{\gamma}_m\ell), \tag{A.100}$$

for any $(j,k) \in [1,2,...,d]^2$.We give the proof in following.

Proof. Based on the sandwich inequality in Theorem 4.10, for any $g \in B_{\mathcal{H}}(1)$, any $\sigma \geqslant \gamma_m$, when $\|g\|_2 \geqslant 2\sigma \geqslant \gamma_m$, $\|g\|_T \geqslant \frac{\|g\|_2}{2} \geqslant \sigma$. Therefore, for any $\sigma \geqslant \gamma_m$,

$$\text{if } \|g\|_T \leqslant \sigma \text{ then } \|g\|_2 \leqslant 2\sigma. \tag{A.101}$$

Using this fact, we proceed the proof in two cases.
Case 1: If $\ell \leqslant \tilde{\gamma}_m$, then

$$\begin{aligned} Z_{T,j,k}(w;\ell) &= \left| \sup_{\|f_{j,k}\|_T \leqslant \ell, \|f_{j,k}\|_{\mathcal{H}} \leqslant 1} \frac{1}{T}\sum_{t=1}^{T} f_{j,k}(X_t)w_{t,j} \right| \\ &\leqslant \left| \sup_{\|f_{j,k}\|_T \leqslant \tilde{\gamma}_m, \|f_{j,k}\|_{\mathcal{H}} \leqslant 1} \frac{1}{T}\sum_{t=1}^{T} f_{j,k}(X_t)w_{t,j} \right|. \end{aligned}$$

Since $\tilde{\gamma}_m \geqslant \gamma_m$, using the fact (A.101), we get

$$Z_{T,j,k}(w;\ell) \leqslant \left| \sup_{\|f_{j,k}\|_2 \leqslant 2\tilde{\gamma}_m, \|f_{j,k}\|_{\mathcal{H}} \leqslant 1} \frac{1}{T} \sum_{t=1}^{T} f_{j,k}(X_t) w_{t,j} \right|.$$

Further, since $\tilde{\gamma}_m \geqslant \tilde{\epsilon}_m$, we are able to use Theorem 4.9 and show that

$$Z_{T,j,k}(w;\ell) \leqslant 4\sqrt{2}\sqrt{\frac{m}{T}}\tilde{\gamma}_m^2.$$

Case 2: If $\ell \geqslant \tilde{\gamma}_m$, we use scaling on f to transform it to Case 1, hence we can show a bound in following.

$$\begin{aligned}
&Z_{T,j,k}(w;\ell) \\
&= \\
&\left| \sup_{\|f_{j,k}\|_T \leqslant \ell, \|f_{j,k}\|_{\mathcal{H}} \leqslant 1} \frac{1}{T} \sum_{t=1}^{T} f_{j,k}(X_t) w_{t,j} \right| \\
&= \\
&\left| \frac{\ell}{\tilde{\gamma}_m} \sup_{\|\frac{\tilde{\gamma}_m}{\ell} f_{j,k}\|_T \leqslant \tilde{\gamma}_m, \|\frac{\tilde{\gamma}_m}{\ell} f_{j,k}\|_{\mathcal{H}} \leqslant \frac{\tilde{\gamma}_m}{\ell}} \frac{1}{T} \sum_{t=1}^{T} \frac{\tilde{\gamma}_m}{\ell} f_{j,k}(X_t) w_{t,j} \right| \\
&\leqslant \\
&\left| \frac{\ell}{\tilde{\gamma}_m} \sup_{\|\tilde{f}_{j,k}\|_T \leqslant \tilde{\gamma}_m, \|\tilde{f}_{j,k}\|_{\mathcal{H}} \leqslant 1} \frac{1}{T} \sum_{t=1}^{T} \tilde{f}_{j,k}(X_t) w_{t,j} \right| \\
&\leqslant \\
&4\sqrt{2}\sqrt{\frac{m}{T}} \ell \tilde{\gamma}_m.
\end{aligned}$$

Therefore, statement (A.100) is true. □

Next, we use proof by contradiction to prove (A.99). If (A.99) fails for a function $f_{j,k}^0$, we can assume $\|f_{j,k}^0\|_{\mathcal{H}} = 1$, otherwise, statement also fails for $\frac{f_{j,k}^0}{\|f_{j,k}^0\|_{\mathcal{H}}}$. Then we

let $\ell = \|f^0_{j,k}\|_T$. Now $\|f^0_{j,k}\|_T \leqslant \ell$, $\|f^0_{j,k}\|_{\mathcal{H}} \leqslant 1$, but

$$\begin{aligned}
&\left|\frac{1}{T}\sum_{t=1}^{T} f^0_{j,k}(X_t)w_{t,j}\right| \\
&\geqslant 4\sqrt{2}\sqrt{\frac{m}{T}}(\tilde{\gamma}^2_m\|f^0_{j,k}\|_H + \tilde{\gamma}_m\|f^0_{j,k}\|_T) \\
&= 4\sqrt{2}\sqrt{\frac{m}{T}}(\tilde{\gamma}^2_m + \tilde{\gamma}_m\ell),
\end{aligned}$$

which contradicts (A.100). Therefore, (A.99) is true. □

Proof of Lemma 4.12. First, using Theorem 4.10, on event $\mathcal{B}_{m,T}$ for any $(j,k) \in [1,2,...,d]^2$,

$$\|f_{j,k}\|_T \leqslant \|f_{j,k}\|_2 + \frac{\gamma_m}{2}, \tag{A.102}$$

for all $f_{j,k} \in B_{\mathcal{H}}(1)$ and $\|f_{j,k}\|_2 \leqslant \gamma_m$. On the other hand, if $\|f_{j,k}\|_2 > \gamma_m$, then the sandwich relation in Theorem 4.10 implies that $\|f_{j,k}\|_T \leqslant 2\|f_{j,k}\|_2$. Therefore, we have

$$\|f_{j,k}\|_T \leqslant 2\|f_{j,k}\|_2 + \frac{\gamma_m}{2} \text{ for all } f_{j,k} \in B_{\mathcal{H}}(1).$$

The proof is completed by noticing $g_{j,k} = 2f_{j,k}$. □

Proof of Lemma 4.13. First, we point out that we only need to show that

$$\|g_j\|_T \geqslant \delta_{m,j}/2 \text{ for all } g_j \in 2\mathcal{F}_j \text{ with } \|g_j\|_2 = \delta_{m,j},$$

because if $\|g_j\|_2 \geqslant \delta_{m,j}$, we can scale g_j to $\frac{\delta_{m,j}}{\|g_j\|_2}g_j$, which belongs to $2\mathcal{F}_j$ as well since $\frac{\delta_{m,j}}{\|g_j\|_2} < 1$. We choose a truncation level $\tau > 0$ and define the function

$$\ell_\tau(u) = \begin{cases} u^2 & \text{if } |u| \leqslant \tau \\ \tau^2 & \text{otherwise} \end{cases}.$$

Since $u^2 \geqslant \ell_\tau(u)$ for all $u \in \mathbb{R}$, we have

$$\frac{1}{T}\sum_{t=1}^{T} g_j^2(X_t) \geqslant \frac{1}{T}\sum_{t=1}^{T} \ell_\tau(g_j(X_t)).$$

The remainder of the proof consists of the following steps:

(1) First, we show that for all $g_j \in 2\mathcal{F}$ with $\|g_j\| = \delta_{m,j}$, we have

$$\mathbb{E}[\ell_\tau(g_j(x))] \geqslant \frac{1}{2}\mathbb{E}[g_j^2(x)] = \frac{\delta_{m,j}^2}{2}.$$

(2) Next we prove that

$$\sup_{g_j \in 2\mathcal{F}_j, \|g_j\|_2 \leqslant \delta_{m,j}} \left| \frac{1}{T}\sum_{t=1}^{T} \ell_\tau(g_j(X_t)) - \mathbb{E}[\ell_\tau(g(X_t))] \right| \leqslant \frac{\delta_{m,j}^2}{4}, \tag{A.103}$$

with high probability for β mixing process with $r \geqslant 1/c_0$.

Putting together the pieces, we conclude that for any $g_j \in \mathcal{F}_j$ with $\|g_j\|_2 = \delta_{m,j}$, we have

$$\frac{1}{T}\sum_{t=1}^{T} g_j^2(X_t) \geqslant \frac{1}{T}\sum_{t=1}^{T} \ell_\tau(g_j(X_t)) \geqslant \frac{\delta_{m,j}^2}{2} - \frac{\delta_{m,j}^2}{4} = \frac{\delta_{m,j}^2}{4},$$

with high probability (to be specified later). This shows that event $\mathcal{D}_{m,T}$ holds with high probability, thereby completing the proof. It remains to establish the claims.
Part 1. Establishing the lower bound for $\mathbb{E}[\ell_\tau(g_j(x))]$:

Proof. We can not use the same proofs as in the independent case from Raskutti et al. (2012), since each element from the multivariate variable $x = (x_1, ..., x_d)$ is not independent from others in the stationary distribution. That is the reason why we need to have Assumption 4. In the independent case, Assumption 4 is shown to be

true in Raskutti et al. (2012). Note that

$$g_j(x) = \sum_{k \in U} g_{j,k}(x_j),$$

for a subset U of cardinality at most $2s_j$, we have

$$\begin{aligned}\mathbb{E}[\ell_\tau(g_j(x))] &\geqslant E[g_j^2(x) I[|g_j(x)| \leqslant \tau]] \\ &= \delta_{m,j}^2 - E[g_j^2(x) I[|g_j(x)| \geqslant \tau]].\end{aligned}$$

Using Cauchy-Schwarz inequality and Markov inequality, we can show that

$$\begin{aligned}(\mathbb{E}[g_j^2(x) I[|g_j(x)| \geqslant \tau]])^2 &\leqslant \mathbb{E}[g_j^4(x)] P(|g_j(x)| \geqslant \tau) \\ &\leqslant \mathbb{E}[g_j^4(x)] \frac{\delta_{m,j}^2}{\tau^2}.\end{aligned}$$

Since $E[g_j^4(x)] \leqslant C\delta_{m,j}^2 = CE[g_j^2(x)]$ given by Assumption 4, by choosing $\tau \geqslant 2\sqrt{C}$, we are able to show that

$$\mathbb{E}[\ell_\tau(g_j(x))] \geqslant \frac{1}{2}\mathbb{E}[g_j^2(x)] = \frac{\delta_{m,j}^2}{2}.$$

□

Part 2. Establishing the probability bound on

$$\left\{ \sup_{g_j \in 2\mathcal{F}_j, \|g_j\|_2 \leqslant \delta_{m,j}} \left| \frac{1}{T} \sum_{t=1}^{T} \ell_\tau(g_j(X_t)) - \mathbb{E}[\ell_\tau(g_j(X_1))] \right| \leqslant \frac{\delta_{m,j}^2}{4} \right\}.$$

Proof. Similar as the proof of Lemma. 4.11, we base our proof on the independent

result from Lemma 4 in Raskutti et al. (2012), which is

$$\begin{aligned}
&\mathbb{P}_0\Bigg(\sup_{g_j\in 2\mathcal{F}_j, \|g_j\|_2\leqslant \delta_{m,j}} \\
&\qquad \left|\frac{1}{m}\sum_{a=1}^{m}\ell_\tau(g(X_a)) - \mathsf{E}[\ell_\tau(g(X_1))]\right| \frac{\delta_{m,j}^2}{12}\Bigg) \\
&\leqslant \tilde{c}_1 \exp(-\tilde{c}_2 m\delta_{m,j}^2).
\end{aligned} \tag{A.104}$$

We let $T = m\ell$. Using the same facts and results as in the proof for Theorem 4.10, we have

$$\begin{aligned}
&\mathbb{P}\Bigg(\frac{\delta_{m,j}^2}{4} \leqslant \sup_{g_j\in 2\mathcal{F}_j, \|g_j\|_2\leqslant \delta_{m,j}} \\
&\qquad \left|\frac{1}{T}\sum_{t=1}^{T}\ell_\tau(g_j(X_t)) - \mathbb{E}[\ell_\tau(g_j(X_1))]\right|\Bigg) \\
&= \mathbb{P}\Bigg(\frac{\delta_{m,j}^2}{4} \leqslant \sup_{g_j\in 2\mathcal{F}_j, \|g_j\|_2\leqslant \delta_{m,j}} \\
&\qquad \left|\frac{1}{\ell}\sum_{b=1}^{\ell}\frac{1}{m}\sum_{a=1}^{m}\ell_\tau(g_j(X_{a,b})) - \mathbb{E}[\ell_\tau(g(X_{a,1}))]\right|\Bigg) \\
&\leqslant \ell P_0\Bigg(\frac{\delta_{m,j}^2}{4} \leqslant \sup_{g_j\in 2\mathcal{F}_j, \|g_j\|_2\leqslant \delta_{m,j}} \\
&\qquad \left|\frac{1}{m}\sum_{a=1}^{m}\ell_\tau(g_j(X_{a,\ell})) - \mathbb{E}[\ell_\tau(g(X_{a,\ell}))]\right|\Bigg) + T\beta(\ell).
\end{aligned}$$

Using (A.104), we conclude that it is upper bounded by

$$\ell\tilde{c}_1\exp(-\tilde{c}_2 m\delta_{m,j}^2) + T\left(\frac{T}{m}\right)^{-r_\beta},$$

which is then upper bounded by $\tilde{c}_3\exp(-\tilde{c}_4 m\delta_{m,j}^2) + T^{-\frac{1-c_0}{c_0}}$ for constants $\tilde{c}_3, \tilde{c}_4$. □

Now, we proved that all claims are correct. Therefore, we complete the proof. □

Proof of Lemma A.26. For ϕ-mixing process with $0.781 \leqslant \phi \leqslant 2$, we can use the

concentration inequality from Kontorovich et al. (2008) to show sharper rate in Lemma A.26 than Theorem 4.10. That concentration inequality is presented in following.

Lemma A.28 (McDirmaid inequality in Kontorovich et al. (2008); Mohri and Rostamizadeh (2010))**.** *Suppose $\mathbb{S}$ is a countable space, $\mathbb{F}_{\mathbb{S}}$ is the set of all subsets of $\mathbb{S}^n$, $\mathbb{Q}$ is a probability measure on $(\mathbb{S}^n, \mathbb{F}_{\mathbb{S}})$ and $g : \mathbb{S}^n \to \mathbb{R}$ is a c-Lipschitz function (with respect to the Hamming metric) on $\mathbb{S}^n$ for some $c > 0$. Then for any $y > 0$,*

$$\mathbb{P}(|g(X) - \mathbb{E}g(X)| \geqslant y) \leqslant 2exp\left(-\frac{y^2}{2nc^2(1 + 2\sum_{\ell=1}^{n-1}\phi(\ell))^2}\right).$$

Its original version is for discrete space, which is then generalized to continuous case in Kontorovich (2007). Here, we use its special form for the ϕ-mixing process which is pointed out in Kontorovich (2007) and Mohri and Rostamizadeh (2010).

For our statement, as pointed out in the proof for Theorem 4.10, since we have $\|f_{j,k}\|_\infty \leqslant 1$, it suffices to bound

$$\mathbb{P}\left(\sup_{j,k}\sup_{f_{j,k}\in B_{\mathcal{H}}(1), \|f_{j,k}\|_2 \leqslant \gamma_m} |\|f_{j,k}\|_T^2 - \|f_{j,k}\|_2^2| \geqslant \frac{\gamma_m^2}{4}\right). \tag{A.105}$$

The proofs are based on independent result from Lemma 7 in Raskutti et al. (2012), which shows that there exists constants $(\tilde{c}_1, \tilde{c}_2)$ such that

$$\begin{aligned}&\mathbb{P}_0\left(\sup_{j,k}\sup_{f_{j,k}\in B_H(1), \|f_{j,k}\|_2 \leqslant \gamma_m}\left|\frac{1}{m}\sum_{t=1}^{m} f_{j,k}^2(\tilde{X}_t) - \|f_{j,k}\|_2^2\right| \geqslant \frac{\gamma_m^2}{10}\right)\\ &\leqslant \tilde{c}_1 exp(-\tilde{c}_2 m\gamma_m^2),\end{aligned}$$

where $\{\tilde{X}_t\}_{t=1}^m$ are i.i.d drawn from the stationary distribution of X_t denoted by $\mathbb{P}_0$.

Now, we can use Lemma A.28 to show the sharper rate. Recall that $\|f_{j,k}\|_\infty \leqslant 1$, we define

$$g(X) = \sup_{f_{j,k}\in B_{\mathcal{H}}(1), \|f_{j,k}\|_2 \leqslant \gamma_m} |\|f_{j,k}\|_T^2 - \|f_{j,k}\|_2^2|.$$

Then,

$$\begin{aligned}
&|g(X) - g(Y)| \\
&\leqslant \sup_{f_{j,k} \in B_{\mathbb{H}}(1), \|f_{j,k}\|_2 \leqslant \gamma_m} \left| \frac{1}{T} \sum_{t=1}^{T} f_{j,k}(X_t)^2 - \frac{1}{T} \sum_{t=1}^{T} f_{j,k}(Y_t)^2 \right| \\
&\leqslant \frac{1}{T} \text{dist}_{\mathcal{HM}}(X, Y),
\end{aligned}$$

where $\text{dist}_{\mathcal{HM}}(X, Y)$ means the Hamming metric between X and Y, which equals to how many paired elements are different between X and Y. Thus, we know that $g(X)$ is $\frac{1}{T}$-Lipschitz with respect to the Hamming metric. Therefore, using Lemma A.28, we show that

$$\begin{aligned}
&\mathbb{P}\left(|g(X) - \mathbb{E}g(X)| \geqslant \frac{\gamma_m^2}{8} \right) \\
&\leqslant 2\exp\left(-\frac{T\gamma_m^4}{128(1 + 2\sum_{\ell=1}^{T-1} \phi(\ell))^2} \right).
\end{aligned}$$

Using the fact that $\phi(\ell) = \ell^{-r_\phi}$, we show that probability is bounded by

$$O(\exp(-\min(T^{2r_\phi - 1}, T)\gamma_m^4)).$$

If we use union bound on d^2 terms, that is at most $O(\exp(2\log(d) - \min(T^{2r_\phi - 1}, T)\gamma_m^4))$. Since $0.781 \leqslant r_\phi \leqslant 2$, we show that $T^{2r_\phi - 1}\gamma_m^4 = m^{\frac{(r_\phi + 2)(2r_\phi - 1)}{r_\phi}}\gamma_m^4 = \Omega(m^2\gamma_m^4)$ and $T\gamma_m^4 = m^{\frac{r_\phi + 2}{r_\phi}}\gamma_m^4 = \Omega(m^2\gamma_m^4)$. Therefore, the probability is at most $c_2\exp(-c_3(m\gamma_m^2)^2)$ for some constants (c_2, c_3).

The remaining proof is then to show that $\mathbb{E}g(X) \leqslant \frac{\gamma_m^2}{8}$. In other words, we need to show that for sufficient large m,

$$\mathbb{E} \sup_{f_{j,k} \in B_{\mathcal{H}}(1), \|f_{j,k}\|_2 \leqslant \gamma_m} \left| \frac{1}{T} \sum_{t=1}^{T} f_{j,k}^2(X_t) - \|f_{j,k}\|_2^2 \right| \leqslant \frac{\gamma_m^2}{8} \tag{A.106}$$

First, we use the same fact and results as in the proof for Theorem 4.10 to show

that

$$\begin{aligned}
&\mathbb{E}\left[\sup_{f_{j,k}\in B_{\mathcal{H}}(1),\|f_{j,k}\|_2\leqslant\gamma_m}\left|\frac{1}{T}\sum_{t=1}^{T}f_{j,k}^2(X_t)-\|f_{j,k}\|_2^2\right|\right]\\
&=\\
&\mathbb{E}\left[\sup_{f_{j,k}\in B_{\mathcal{H}}(1),\|f_{j,k}\|_2\leqslant\gamma_m}\left|\frac{1}{\ell}\sum_{b=1}^{\ell}\frac{1}{m}\sum_{a=1}^{m}f_{j,k}^2(X_{a,b})-\|f_{j,k}\|_2^2\right|\right]\\
&\leqslant\\
&\frac{1}{\ell}\sum_{b=1}^{\ell}\mathbb{E}\left[\sup_{f_{j,k}\in B_{\mathcal{H}}(1),\|f_{j,k}\|_2\leqslant\gamma_m}\left|\frac{1}{m}\sum_{a=1}^{m}f_{j,k}^2(X_{a,b})-\|f_{j,k}\|_2^2\right|\right]\\
&=\\
&\mathbb{E}\left[\sup_{f_{j,k}\in B_{\mathcal{H}}(1),\|f_{j,k}\|_2\leqslant\gamma_m}\left|\frac{1}{m}\sum_{a=1}^{m}f_{j,k}^2(X_{a,\ell})-\|f_{j,k}\|_2^2\right|\right].
\end{aligned}$$

Using the fact that $\mathbb{E}[Z]=\mathbb{E}[ZI(Z\leqslant\delta)]+\mathbb{E}[ZI(Z\geqslant\delta)]\leqslant\delta+\|Z\|_\infty\mathbb{P}(Z\geqslant\delta)$ and $\|f_{j,k}\|_\infty\leqslant 1$, we show an upper bound

$$\begin{aligned}
&\delta+\\
&2\mathbb{P}\left(\sup_{f_{j,k}\in B_{\mathcal{H}}(1),\|f_{j,k}\|_2\leqslant\gamma_m}\left|\frac{1}{m}\sum_{a=1}^{m}f_{j,k}^2(X_{a,\ell})-\|f_{j,k}\|_2^2\right|\geqslant\delta\right),
\end{aligned}$$

for any $\delta>0$. As in the proof of Theorem 4.10, we use Lemma 2 in Nobel and Dembo (1993) to connect the dependence probability with independence probability, which gives us

$$\begin{aligned}
&\delta+2\mathbb{P}\left(\sup_{f_{j,k}\in B_{\mathcal{H}}(1),\|f_{j,k}\|_2\leqslant\gamma_m}\left|\|f_{j,k}\|_m^2-\|f_{j,k}\|_2^2\right|\geqslant\delta\right)\\
&\leqslant\delta+m\phi(\ell)+\\
&\quad P_0\left(\sup_{f_{j,k}\in B_{\mathcal{H}}(1),\|f_{j,k}\|_2\leqslant t}\left|\|f_{j,k}\|_m^2-\|f_{j,k}\|_2^2\right|\geqslant\delta\right).
\end{aligned}$$

We choose δ to be $\frac{\gamma_m^2}{10}$, then using (A.105), we have the upper bound

$$\frac{\gamma_m^2}{10} + m\phi(\ell) + P_0\left(\sup_{f\in B_H(1),\|f\|_2\leqslant t}\left|\|f\|_m^2 - \|f\|_2^2\right| \geqslant \frac{\gamma_m^2}{10}\right)$$
$$\leqslant \frac{\gamma_m^2}{10} + m\phi(\ell) + \tilde{c}_1\exp(-\tilde{c}_2 m\gamma_m^2).$$

We require $m\gamma_m^2 = \Omega(-\log(\gamma_m))$, which is the same as Raskutti et al. (2012). Based on our assumptions, $m\phi(\ell) = m(m^{2/r_\phi})^{-r_\phi} = m^{-1} = o(\gamma_m^2)$ since $m\gamma_m^2 \to \infty$ as $m \to \infty$. Therefore, for sufficiently large m, that expectation is bounded by $\frac{\gamma_m^2}{8}$. That completes the proof.

For the follow-up statement, condition on event $\mathcal{B}_{m,T}$, for any $g_{j,k} \in B_{\mathcal{H}}(1)$ with $\|g_{j,k}\|_2 \geqslant \gamma_m$, we have $h_{j,k} = \gamma_m \frac{g_{j,k}}{\|g_{j,k}\|_2}$ is in $B_{\mathcal{H}}(1)$ and $\|h_{j,k}\|_2 \leqslant \gamma_m$. Therefore, we have

$$\left|\left\|\gamma_m \frac{g_{j,k}}{\|g_{j,k}\|_2}\right\|_T - \left\|\gamma_m \frac{g_{j,k}}{\|g_{j,k}\|_2}\right\|_2\right| \leqslant \frac{\gamma_m}{2},$$

which implies

$$|\|g_{j,k}\|_T - \|g_{j,k}\|_2| \leqslant \frac{1}{2}\|g_{j,k}\|_2.$$

□

Proof of Lemma A.27. We follow the outline of proof for Lemma 4.13. The only difference is here is the proof for showing

$$\sup_{g_j\in 2\mathcal{F}_j,\|g_j\|_2\leqslant\delta_{m,j}}\left|\frac{1}{T}\sum_{t=1}^{T}\ell_\tau(g_j(X_t)) - \mathbb{E}[\ell_\tau(g(X_t))]\right| \leqslant \frac{\delta_{m,j}^2}{4},$$

with high probability for ϕ mixing process with $0.781 \leqslant r \leqslant 2$.

To show that, we use Lemma A.28 as in the proof of Lemma A.26 and define

$$h(X) = \sup_{g_j\in 2\mathcal{F}_j,\|g_j\|_2\leqslant\delta_{m,j}}\left|\frac{1}{T}\sum_{t=1}^{T}\ell_\tau(g_j(X_t)) - \mathbb{E}[\ell_\tau(g_j(X_1))]\right|.$$

We have

$$
\begin{aligned}
&|h(X) - h(Y)| \\
&\leqslant \sup_{g_j \in 2\mathcal{F}_j, \|g_j\|_2 \leqslant \delta_{m,j}} \left| \frac{1}{T} \sum_{t=1}^{T} (\ell_\tau(g_j(X_t)) - \ell_\tau(g_j(Y_t))) \right| \\
&\leqslant \frac{\tau^2}{T} \mathrm{dist}_{\mathcal{HM}}(X, Y),
\end{aligned}
$$

which give us

$$
\begin{aligned}
&\mathbb{P}\left(|h(X) - \mathbb{E}h(X)| \geqslant \frac{\delta_{m,j}^2}{8} \right) \\
&\leqslant O\left(exp\left(-\min(T^{2r_\phi - 1}, T) \frac{\delta_{m,j}^4}{\tau^4} \right) \right) \\
&\leqslant c_2 exp(-c_3 (m\delta_{m,j}^2)^2),
\end{aligned}
$$

following the same analyses as in the proof of Lemma A.26.

As in the proof of Lemma A.26, we then need to show that for sufficient large m,

$$
\begin{aligned}
&E \sup_{g_j \in 2\mathcal{F}_j, \|g_j\|_2 \leqslant \delta_{m,j}} \left| \frac{1}{T} \sum_{t=1}^{T} \ell_\tau(g_j(X_t)) - \mathbb{E}[\ell_\tau(g_j(X))] \right| \\
&\leqslant \frac{\delta_{m,j}^2}{8}.
\end{aligned}
$$

Using the same facts and results as we mentioned in the proof of Theorem 4.10

and Lemma A.26, we show the upper bound in following.

$$
\begin{aligned}
&\mathbb{E} \sup_{g_j \in 2\mathcal{F}_j, \|g_j\|_2 \leqslant \delta_{m,j}} \left| \frac{1}{T} \sum_{t=1}^{T} \ell_\tau(g_j(X_t)) - \mathbb{E}[\ell_\tau(g_j(X_1))] \right| \\
&\leqslant \\
&\mathbb{E} \sup_{g_j \in 2\mathcal{F}_j, \|g_j\|_2 \leqslant \delta_{m,j}} \left| \frac{1}{m} \sum_{a=1}^{m} \ell_\tau(g_j(X_{a,\ell})) - \mathbb{E}[\ell_\tau(g_j(X_{a,\ell}))] \right| \\
&\leqslant \\
&\delta + 2\tau^2 P\Bigg(\sup_{g_j \in 2F_j, \|g_j\|_2 \leqslant \delta_{m,j}} \\
&\qquad \left| \frac{1}{m} \sum_{a=1}^{m} \ell_\tau(g_j(X_{a,\ell})) - \mathbb{E}[\ell_\tau(g_j(X))] \right| \geqslant \delta \Bigg) \\
&\leqslant \\
&\delta + 2\tau^2 m\phi(\ell) + 2\tau^2 \mathbb{P}_0\Bigg(\sup_{g_j \in 2\mathcal{F}_j, \|g_j\|_2 \leqslant \delta_{m,j}} \\
&\qquad \left| \frac{1}{m} \sum_{a=1}^{m} \ell_\tau(g_j(X_{a,\ell})) - \mathbb{E}[\ell_\tau(g_j(X))] \right| \geqslant \delta \Bigg) \\
&\leqslant \\
&\frac{\delta_{m,j}^2}{12} + 2\tau^2 m\phi(\ell) + 2\tau^2 \mathbb{P}_0\Bigg(\sup_{g_j \in 2\mathcal{F}_j, \|g_j\|_2 \leqslant \delta_m} \\
&\qquad \left| \frac{1}{m} \sum_{a=1}^{m} \ell_\tau(g_j(X_{a,\ell})) - \mathbb{E}[\ell_\tau(g_j(X))] \right| \geqslant \frac{\delta_{m,j}^2}{12} \Bigg) \\
&\leqslant \\
&\frac{\delta_{m,j}^2}{12} + 2\tau^2 m\phi(\ell) + 2\tau^2 \tilde{c}_1 \exp -(\tilde{c}_2 m \delta_{m,j}^2),
\end{aligned}
$$

which is bounded by $\frac{\delta_{m,j}^2}{8}$ for sufficiently large m, using similar arguments as in the proof for Lemma 4.11. That completes the proof. □

A.4 Supplement for Chapter 5

In this section, we first discuss the extensions of the model to multiple outputs and classification and discuss how to incorporate the bias terms in NN. Then we show the proofs for the theorems in the main body. Finally, we present details for models used in experiments and provide more experiment results.

The extension to multiple outputs, classification and including bias terms

For the output $\mathbf{y} \in \mathbb{R}^d$, we permit our computation skeleton to have d output nodes as well. Then after we run our construction algorithm, we obtain a BNN with d outputs $\mathbf{f}^L$ at the last layer. Then the analysis and properties for the single output case also hold for the multiple output case.

In regression task, we assume the likelihood $p(\mathbf{y}|\mathbf{F}^L)$ of the output $\mathbf{y}$ to be a normal distribution, given the input matrix $\mathbf{X}$ with n samples and the relevant output $\mathbf{F}^L$ at the last layer of BNN. We output $\mathbf{F}^L$ to estimate the mean of $\mathbf{y}$. The relevant loss in the optimization of ELBO is the mean square loss. This is usually considered for the regression task. In a classification task with $\mathbf{y} \in \{0, 1, ..., k\}$ in k categories, we assume the likelihood $p(y_i|\mathbf{F}^L_{i\cdot}) = \frac{\exp(\mathbf{F}^L_{iy_i})}{\sum_{j=1}^k \exp(\mathbf{F}^L_{ij})}$ for $1 \leqslant i \leqslant n$. Then the relevant loss in the optimization of ELBO is

$$\frac{1}{n}\sum_{i=1}^{n} \log\left(\frac{\exp(\mathbf{F}^L_{iy_i})}{\sum_{j=1}^k \exp(\mathbf{F}^L_{ij})}\right).$$

We can add bias terms into the framework. The bias term with random weight can either be incorporated into the construction of RB Daniely et al. (2016) or be treated as a parameter in the BNN Gal and Ghahramani (2016). We refer one to see these two works for incorporating bias terms.

Remark. The statements for the extension to multiple outputs and classification hold for DGPs as well.

Proofs for theorems in the main body

In this section, we give the proofs for theorems in the main body.

The proof for the relation between activation functions and kernels

First, we prove theorems on the relation between activation functions and kernels.

The uniform concentration bound for C-bounded activation functions and its proof

First, we present the uniform concentration bound for C-bounded activation functions and its proof.

Theorem 0. If the activation function $\sigma_{\mathcal{K}}$ is C-bounded, meaning it is continuously differentiable and $\|\sigma_{\mathcal{K}}\|_\infty, \|\sigma'_{\mathcal{K}}\|_\infty \leqslant C$, then for every $1 < \ell \leqslant L$, on a compact set $\mathcal{M} \in \mathbb{R}^d$ with diameter $diam(\mathcal{M})$, with probability at least $1 - c_1 diam(\mathcal{M})^2 \exp\left\{-\frac{\epsilon^2 r}{8(1+d)C}\right\}$,

$$\sup_{\sigma(\mathbf{f}^{\ell-1}(\mathbf{x})),\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'))\in\mathcal{M}} |\hat{\mathcal{K}}^\ell(\mathbf{f}^{\ell-1}(\mathbf{x}),\mathbf{f}^{\ell-1}(\mathbf{x}')) - \mathcal{K}^\ell(\mathbf{f}^{\ell-1}(\mathbf{x}),\mathbf{f}^{\ell-1}(\mathbf{x}'))| \leqslant \epsilon,$$

for a constant $c_1 > 0$.

Proof. (a) For a C-bounded activation function, since $\|\sigma_{\mathcal{K}}(\cdot)\|_\infty \leqslant C$, for fixed $\mathbf{f}^{\ell-1}(\mathbf{x})$ and $\mathbf{f}^{\ell-1}(\mathbf{x}')$, the r random variables $\{\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))^\top \mathbf{w}_i)\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'))^\top \mathbf{w}_i)\}_{i=1}^r$ are independent and lie in a bounded interval $[-C, C]$. Then using Hoeffdings' inequality, we get that

$$\mathbb{P}(|\hat{\mathcal{K}}^\ell(\mathbf{f}^{\ell-1}(\mathbf{x}),\mathbf{f}^{\ell-1}(\mathbf{x}')) - \mathcal{K}^\ell(\mathbf{f}^{\ell-1}(\mathbf{x}),\mathbf{f}^{\ell-1}(\mathbf{x}'))| \geqslant \epsilon) \leqslant 2\exp(-\frac{2r\epsilon^2}{4C^2}). \quad \text{(A.107)}$$

Next we show that for a compact set $\mathcal{M}$ of $\mathbb{R}^d$ with diameter $diam(\mathcal{M})$, with probability at least $1 - 2^{11}\left(\frac{C^4 d\ diam(\mathcal{M})^2}{\epsilon^2 r}\right)^{\frac{d}{1+d}} \exp\left\{-\frac{r\epsilon^2}{8(1+d)C^2}\right\}$,

$$\sup_{\sigma(\mathbf{f}^{\ell-1}(\mathbf{x})),\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'))\in\mathcal{M}} |\hat{\mathcal{K}}^{\ell}(\mathbf{f}^{\ell-1}(\mathbf{x}),\mathbf{f}^{\ell-1}(\mathbf{x}')) - \mathcal{K}^{\ell}(\mathbf{f}^{\ell-1}(\mathbf{x}),\mathbf{f}^{\ell-1}(\mathbf{x}'))| \leqslant \epsilon.$$

Since $\mathcal{M}$ has diameter $\mathtt{diam}(\mathcal{M})$, we can find δ-net that covers $\mathcal{M}$ using at most $T = (4\mathtt{diam}(\mathcal{M})/\delta)^{d}$ balls of radius δ. Let $\{\Delta_i\}_{i=1}^{T}$ denote the centers of these balls. Then using (A.107) and union bounds, for any two centers, such as Δ_1 and Δ_2, with probability at least $1 - 2\exp(\log(T^2) - \frac{2r\epsilon^2}{16C^2})$,

$$|\hat{\mathcal{K}}^{\ell}(\Delta_1,\Delta_2) - \mathcal{K}^{\ell}(\Delta_1,\Delta_2)| \leqslant \frac{\epsilon}{2}. \tag{A.108}$$

For the function $\mathbf{u}(\mathbf{x},\mathbf{x}') = \hat{\mathcal{K}}^{\ell}(\mathbf{f}^{\ell-1}(\mathbf{x}),\mathbf{f}^{\ell-1}(\mathbf{x}')) - \mathcal{K}^{\ell}(\mathbf{f}^{\ell-1}(\mathbf{x}),\mathbf{f}^{\ell-1}(\mathbf{x}'))$, we have the inequality from partial derivative that

$$|\mathbf{u}(\mathbf{x},\mathbf{x}') - \mathbf{u}(\mathbf{x}_0,\mathbf{x}_0')| \leqslant L_{\sigma_{\mathcal{K}}}(||\sigma(\mathbf{f}^{\ell-1}(\mathbf{x})) - \sigma(\mathbf{f}^{\ell-1}(\mathbf{x}_0))||_2 + ||\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}')) - \sigma(\mathbf{f}^{\ell-1}(\mathbf{x}_0'))||_2), \tag{A.109}$$

where

$$\begin{aligned}
L_{\sigma_{\mathcal{K}}} = \arg \max_{\sigma(\mathbf{f}^{\ell-1}(\mathbf{x})),\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'))\in\mathcal{M}} ||&\frac{1}{r}\sum_{i=1}^{r} \frac{\partial\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))^{T}\mathbf{w}_i)}{\partial\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))}\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'))^{T}\mathbf{w}_i) \\
&- \mathbb{E}_{\mathbf{w}}\frac{\partial\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))^{T}\mathbf{w})}{\partial\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))}\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'))^{T}\mathbf{w})||_2 \\
= ||&\frac{1}{r}\sum_{i=1}^{r} \frac{\partial\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}^*))^{T}\mathbf{w}_i)}{\partial\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))}\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'^*))^{T}\mathbf{w}_i) \\
&- \mathbb{E}_{\mathbf{w}}\frac{\partial\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}^*))^{T}\mathbf{w})}{\partial\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))}\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'^*))^{T}\mathbf{w})||_2.
\end{aligned}$$

We also have that

$$
\begin{aligned}
\mathbb{E}L_{\sigma_{\mathcal{K}}}^2 &= \mathbb{E}\|\frac{1}{r}\sum_{i=1}^{r}\frac{\partial\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}^*))^T\mathbf{w}_i)}{\partial\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))}\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'^*))^T\mathbf{w}_i)\\
&- \mathbb{E}_{\mathbf{w}}\frac{\partial\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}^*))^T\mathbf{w})}{\partial\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))}\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'^*))^T\mathbf{w})\|_2^2\\
&= \mathbb{E}\|\frac{1}{r}\sum_{i=1}^{r}\frac{\partial\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}^*))^T\mathbf{w}_i)}{\partial\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))}\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'^*))^T\mathbf{w}_i)\|_2^2\\
&- \|\mathbb{E}_{\mathbf{w}}\frac{\partial\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}^*))^T\mathbf{w})}{\partial\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))}\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'^*))^T\mathbf{w})\|_2^2\\
&= \frac{1}{r^2}\sum_{i=1}^{r}\mathbb{E}\|\frac{\partial\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}^*))^T\mathbf{w}_i)}{\partial\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))}\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'^*))^T\mathbf{w}_i)\|_2^2\\
&- \frac{1}{r^2}\|\mathbb{E}_{\mathbf{w}}\frac{\partial\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}^*))^T\mathbf{w})}{\partial\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))}\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'^*))^T\mathbf{w})\|_2^2\\
&\leqslant \frac{1}{r^2}C^4\sum_{i=1}^{r}\mathbb{E}\|\mathbf{w}_i\|_2^2\\
&= \frac{C^4 d}{r}.
\end{aligned}
$$

Therefore, by Markov's inequality,

$$
\mathbb{P}(L_{\sigma_{\mathcal{K}}} \geqslant \frac{\epsilon}{4\delta}) \leqslant \mathbb{E}L_{\sigma_{\mathcal{K}}}^2\frac{16\delta^2}{\epsilon^2} \leqslant \frac{16\delta^2 C^4 d}{\epsilon^2 r}
$$

Then using Eq. (A.109), with probability at least $1 - \frac{16\delta^2 C^4 d}{\epsilon^2 r}$,

$$
|u(\mathbf{x},\mathbf{x}') - u(\mathbf{x}_0,\mathbf{x}_0')| \leqslant \frac{\epsilon}{2}
$$

This inequality combined with Eq. (A.108) enables us to conclude that

$$
\sup_{\sigma(\mathbf{f}^{\ell-1}(\mathbf{x})),\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'))\in\mathcal{M}} |\hat{\mathcal{K}}^{\ell}(\mathbf{f}^{\ell-1}(\mathbf{x}),\mathbf{f}^{\ell-1}(\mathbf{x}')) - \mathcal{K}^{\ell}(\mathbf{f}^{\ell-1}(\mathbf{x}),\mathbf{f}^{\ell-1}(\mathbf{x}'))| \leqslant \epsilon.
$$

with probability at least $1 - \frac{16\delta^2 C^4 d}{\epsilon^2 r} - 2\exp(\log(T^2) - \frac{2r\epsilon^2}{16C^2})$. Recall that $T = (4\mathtt{diam}(\mathcal{M})/\delta)^d$, so the probability has a format of $1 - \kappa_1\delta^2 - \kappa_2\delta^{-2d}$ for δ. By setting $\delta = \frac{\kappa_2}{\kappa_1}^{\frac{1}{2+2d}}$, we have the probability as $1 - 2\kappa_1^{\frac{2d}{2+2d}}\kappa_2^{\frac{2}{2+2d}}$. So the probability is at least

$$1 - 2^{11}\left(\frac{C^4 d\ \mathtt{diam}(\mathcal{M})^2}{\epsilon^2 r}\right)^{\frac{d}{1+d}} \exp\left\{-\frac{r\epsilon^2}{8(1+d)C^2}\right\}$$

□

Proof of Theorem 1 for ReLU

We have seen how to control the distance between empirical kernel and the expectation kernel uniformly for C-bounded activation functions, now we present the proof for ReLU activation functions.

Theorem A.29. *If the activation function $\sigma_{\mathcal{K}}$ is ReLU, then for every $1 \leqslant \ell \leqslant L$, on a compact set $\mathcal{M} \in \mathbb{R}^d$ with diameter $\mathtt{diam}(\mathcal{M})$ and $\max_{\Delta\in\mathcal{M}} \|\Delta\|_2 \leqslant c_{\mathcal{M}}$, with probability at least $1 - c_1 c_{\mathcal{M}} \mathtt{diam}(\mathcal{M})^2 exp\left\{-\frac{r\epsilon^2}{8(1+d)\nu_{\mathcal{M}}^2}\right\}$,*

$$\sup_{\sigma(\mathbf{f}^{\ell-1}(\mathbf{x})),\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'))\in\tilde{\mathcal{M}}} |\hat{\mathcal{K}}^{\ell}(\mathbf{f}^{\ell-1}(\mathbf{x}), \mathbf{f}^{\ell-1}(\mathbf{x}')) - \mathcal{K}^{\ell}(\mathbf{f}^{\ell-1}(\mathbf{x}), \mathbf{f}^{\ell-1}(\mathbf{x}'))| \leqslant \epsilon,$$

for a constant $c_1 > 0$ and a parameter $\nu_{\mathcal{M}}$ depending on $\mathcal{M}$. Here, $\tilde{\mathcal{M}}$ specifies that we require $\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))$ and $\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'))$ to be two vectors in $\mathcal{M}$ that are not collinear.

Proof. For the ReLU activation $\sigma_{\mathcal{K}}(x) = \max(0, x)$, we use concentration bound for sub-exponential random variable to show the result for fixed points. We define $u = \sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))^T\mathbf{w})\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'))^T\mathbf{w})$. Our first goal is to compute $\mathbb{E}_{\mathbf{w}}[e^{\lambda u}]$. Since $\mathbf{w}$ follows a normal distribution that is symmetric, how we choose axis does not influence the results. Therefore, we choose axis such that $\sigma(\mathbf{f}^{\ell-1}(\mathbf{x})) = \mathbf{e}_1\|\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))\|_2$ and $\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}')) = (\mathbf{e}_1\cos\theta + \mathbf{e}_2\sin\theta)\|\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'))\|_2$ where $\mathbf{e}_1$ and $\mathbf{e}_2$ refer to standard vector for the first and second axis. We denote $C_f = \|\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))\|_2\|\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'))\|_2 \leqslant c_{\mathcal{M}}^2$.

$$\mathbb{E}_{\mathbf{w}}[e^{\lambda u}] = \frac{1}{2\pi}\int_{-\infty}^{\infty} dw_1 \int_{-\infty}^{\infty} dw_2 e^{-\frac{1}{2}(w_1^2+w_2^2)} e^{\lambda C_f \max(0,w_1)\max(0,w_1\cos\theta + w_2\sin\theta)}.$$

We switch (w_1, w_2) by $(\tilde{w}_1, \tilde{w}_2) = (w_1, w_1\cos\theta + w_2\sin\theta)$, then we get that

$$\mathbb{E}_{\mathbf{w}}[e^{\lambda u}] = \frac{1}{2\pi\sin\theta}\int_0^{\infty} d\tilde{w}_1 \int_0^{\infty} d\tilde{w}_2 e^{-\frac{\tilde{w}_1^2+\tilde{w}_2^2-2\tilde{w}_1\tilde{w}_2\cos\theta}{2\sin\theta^2}} e^{\lambda C_f \tilde{w}_1\tilde{w}_2}.$$

We switch $(\tilde{w}_1, \tilde{w}_2)$ by $(\tilde{r}, \tilde{\phi})$ with $\tilde{w}_1 = \tilde{r}\sin\tilde{\phi}$ and $\tilde{w}_2 = \tilde{r}\cos\tilde{\phi}$, then we get that

$$\mathbb{E}_{\mathbf{w}}[e^{\lambda u}] = \frac{1}{2\pi\sin\theta}\int_0^{\pi/2} d\tilde{\phi} \int_0^{\infty} \tilde{r} d\tilde{r} e^{-\tilde{r}^2\frac{1-\sin 2\tilde{\phi}\cos\theta}{2\sin\theta^2}} e^{\tilde{r}^2\frac{\lambda C_f \sin 2\tilde{\phi}}{2}}.$$

Through the known mean calculation of half normal distribution that

$$\frac{a\sqrt{2}}{\sqrt{\pi}} = \int_{x\geqslant 0} x dx \frac{\sqrt{2}}{a\sqrt{\pi}} \exp(-\frac{x^2}{2a^2}),$$

for any a, we know that

$$a^2 = \int_{x\geqslant 0} x dx \exp(-\frac{x^2}{2a^2}),$$

for any a. We use this relation to calculate the integral of $\tilde{r}$ and we get that

$$\begin{aligned}\mathbb{E}_{\mathbf{w}}[e^{\lambda u}] &= \frac{1}{2\pi\sin\theta}\int_0^{\pi/2} d\tilde{\phi}\frac{\sin\theta^2}{1-\sin 2\tilde{\phi}\cos\theta - \lambda C_f \sin 2\tilde{\phi}\sin\theta^2} \\ &= \frac{\sin\theta}{2\pi}\int_0^{\pi/2} d\alpha \frac{1}{1-\cos\alpha\cos\theta - \lambda C_f \cos\alpha\sin\theta^2},\end{aligned}$$

by switching $\tilde{\phi}$ to $\alpha = 2\tilde{\phi} - \frac{\pi}{2}$. It is known that

$$\int_0^{\xi} d\alpha \frac{1}{1-\cos\alpha\cos\theta} = \frac{1}{\sin\theta}\tan^{-1}\left(\frac{\sin\theta\sin\xi}{\cos\xi-\cos\theta}\right),$$

which can be verified by calculating the derivative of the right side Cho and Saul (2009). Therefore, by setting $\xi = \frac{\pi}{2}$,

$$\int_0^{\pi} 2d\alpha \frac{1}{1-\cos\alpha\cos\theta} = \frac{\pi-\theta}{\sin\theta}.$$

We define $\gamma = \arccos(\cos\theta + \lambda C_f \sin\theta^2)$ for $0 \leqslant \gamma \leqslant \pi$ under the requirement that

$$-\frac{1+\cos\theta}{C_f\sin\theta^2} \leqslant \lambda \leqslant \frac{1-\cos\theta}{C_f\sin\theta^2}.$$

Then we get that

$$\mathbb{E}_{\mathbf{w}}[e^{\lambda u}] = \frac{\sin\theta}{2\pi}\frac{\pi-\gamma}{\sin\gamma}.$$

Since $0 \leqslant \gamma, \theta \leqslant \pi$, now we further assume that $-\frac{b}{C\sin\theta^2} \leqslant \lambda \leqslant \frac{b}{C\sin\theta^2}$, then $\cos\theta - b \leqslant \cos\gamma \leqslant \cos\theta + b$. For a enough small b, we have that

$$\mathbb{E}_{\mathbf{w}}[e^{\lambda u}] \leqslant \frac{3(\pi-\theta)}{4\pi}. \tag{A.110}$$

From Cho and Saul (2009),

$$\mathbb{E}_{\mathbf{w}}[\lambda u] = \frac{2\lambda C_f}{\pi}(\sin\theta + \cos\theta(\pi-\theta)).$$

Therefore, we combine it with Eq. (A.110) to get that

$$\mathbb{E}_{\mathbf{w}}[e^{\lambda(u-\mathbb{E}_{\mathbf{w}}[u])}] \leqslant \frac{3(\pi-\theta)}{4\pi}\exp(-\lambda\frac{2C_f(\sin\theta+\cos\theta(\pi-\theta))}{\pi}),$$

for $-\frac{b}{C\sin\theta^2} \leqslant \lambda \leqslant \frac{b}{C\sin\theta^2}$ with a enough small b.

Because for $\frac{\pi}{2} \leqslant \theta \leqslant \pi$, $\cos\theta(\pi-\theta) \geqslant \cos\theta\tan(\pi-\theta) = -\sin\theta \geqslant 0$, we always can define $c = \frac{2C_f(\sin\theta+\cos\theta(\pi-\theta))}{\pi} \geqslant 0$ and it monotonically decreases to zero at $\theta = \pi$. Then we define $\nu^2 = \frac{c^2}{2\log(\frac{4\pi}{3(\pi-\theta)})}$ that can guarantee

$$\mathbb{E}[e^{\lambda(u-\mathbb{E}[u])}] \leqslant \exp(\frac{\nu^2\lambda^2}{2}),$$

for $|\lambda| \leqslant \frac{b}{C\sin\theta^2}$. This means that u follows a sub-exponential distribution and now we can use concentration bound to derive that

$$\mathbb{P}[|u-\mathbb{E}[u]| \geqslant \epsilon] \leqslant \begin{cases} 2e^{-\frac{\epsilon^2}{2\nu^2}} \text{ if } 0 \leqslant \epsilon \leqslant \frac{\nu^2 b}{C\sin\theta^2} \\ 2e^{-\frac{\epsilon b}{2C\sin\theta^2}} \text{ for } \epsilon > \frac{\nu^2 b}{C\sin\theta^2} \end{cases}.$$

Therefore, for a small error ϵ, we can consider that

$$\mathbb{P}[|u-\mathbb{E}[u]| \geqslant \epsilon] \leqslant 2e^{-\frac{\epsilon^2}{2\nu^2}}.$$

The concentration inequality also applied to the average of r independent random variable u_i, which are defined for r independent $\mathbf{w}_i$. It shows that

$$\mathbb{P}[|\frac{1}{r}\sum_{i=1}^{r} u_i - \mathbb{E}[u]| \geqslant \epsilon] \leqslant 2e^{-\frac{r\epsilon^2}{2\nu^2}}. \tag{A.111}$$

Therefore, we obtain the concentration bound for fixed $\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))$ and $\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'))$.

Next, we show the result for a set $\mathcal{M}$. When $||\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))||_2||\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'))||_2$ is bounded and the angle between $\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))$ and $\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'))$ lies in $(0,\pi)$ meaning there is no collinearity, then we have an upper bound $\nu_{\mathcal{M}}$ for ν depending on that two conditions. Therefore, we similar choose T balls with radius δ to cover $\mathcal{M}$ as in the proof for Theorem A.29. Let $\{\Delta_i\}_{i=1}^{T}$ denote the centers of these balls. Then using (A.111) and union bounds, for any two centers, such as Δ_1 and Δ_2, with probability at least $1 - 2\exp\left(\log(T^2) - \frac{r\epsilon^2}{8\nu_{\mathcal{M}}^2}\right)$,

$$|\hat{\mathcal{K}}^{\ell}(\Delta_1,\Delta_2) - \mathcal{K}^{\ell}(\Delta_1,\Delta_2)| \leqslant \frac{\epsilon}{2}. \tag{A.112}$$

Then for the function $\mathbf{u}(\mathbf{x}, \mathbf{x}') = \hat{\mathcal{K}}^{\ell}(\mathbf{f}^{\ell-1}(\mathbf{x}), \mathbf{f}^{\ell-1}(\mathbf{x}')) - \mathcal{K}^{\ell}(\mathbf{f}^{\ell-1}(\mathbf{x}), \mathbf{f}^{\ell-1}(\mathbf{x}'))$, we have the inequality from partial derivative that

$$|\mathbf{u}(\mathbf{x}, \mathbf{x}') - \mathbf{u}(\mathbf{x}_0, \mathbf{x}_0')| \leqslant L_{\sigma_{\mathcal{K}}}(||\sigma(\mathbf{f}^{\ell-1}(\mathbf{x})) - \sigma(\mathbf{f}^{\ell-1}(\mathbf{x}_0))||_2 + ||\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}')) - \sigma(\mathbf{f}^{\ell-1}(\mathbf{x}_0'))||_2), \tag{A.113}$$

where

$$\begin{aligned}
L_{\sigma_{\mathcal{K}}} = \arg \max_{\sigma(\mathbf{f}^{\ell-1}(\mathbf{x})), \sigma(\mathbf{f}^{\ell-1}(\mathbf{x}')) \in \mathcal{M}} & ||\frac{1}{r} \sum_{i=1}^{r} \frac{\partial \sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))^{\mathsf{T}} \mathbf{w}_i)}{\partial \sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))} \sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'))^{\mathsf{T}} \mathbf{w}_i) \\
& - \mathbb{E}_{\mathbf{w}} \frac{\partial \sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))^{\mathsf{T}} \mathbf{w})}{\partial \sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))} \sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'))^{\mathsf{T}} \mathbf{w})||_2 \\
= & ||\frac{1}{r} \sum_{i=1}^{r} \frac{\partial \sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}^*))^{\mathsf{T}} \mathbf{w}_i)}{\partial \sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))} \sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'^*))^{\mathsf{T}} \mathbf{w}_i) \\
& - \mathbb{E}_{\mathbf{w}} \frac{\partial \sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}^*))^{\mathsf{T}} \mathbf{w})}{\partial \sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))} \sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'^*))^{\mathsf{T}} \mathbf{w})||_2.
\end{aligned}$$

We also have that

$$\begin{aligned}
\mathbb{E} L_{\sigma_{\mathcal{K}}}^2 &= \frac{1}{r^2} \sum_{i=1}^{r} \mathbb{E}||\frac{\partial \sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}^*))^{\mathsf{T}} \mathbf{w}_i)}{\partial \sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))} \sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'^*))^{\mathsf{T}} \mathbf{w}_i)||_2^2 \\
&\quad - \frac{1}{r^2} ||\mathbb{E}_{\mathbf{w}} \frac{\partial \sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}^*))^{\mathsf{T}} \mathbf{w})}{\partial \sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))} \sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'^*))^{\mathsf{T}} \mathbf{w})||_2^2 \\
&\leqslant \frac{1}{r^2} \sum_{i=1}^{r} \mathbb{E}||\frac{\partial \sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}^*))^{\mathsf{T}} \mathbf{w}_i)}{\partial \sigma(\mathbf{f}^{\ell-1}(\mathbf{x}))} \sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'^*))^{\mathsf{T}} \mathbf{w}_i)||_2^2.
\end{aligned}$$

Since $\sigma_{\mathcal{K}}$ is ReLU, $||\sigma_{\mathcal{K}}'||_\infty \leqslant 1$, therefore we get that

$$\mathbb{E} L_{\sigma_{\mathcal{K}}}^2 \leqslant \frac{1}{r} \mathbb{E}|\sigma_{\mathcal{K}}(\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'^*))^{\mathsf{T}} \mathbf{w})|||\mathbf{w}||_2^2.$$

Again, since $\mathbf{w}$ follows a normal distribution that is symmetric, we choose axis

to satisfy $\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'^*)) = \mathbf{e}_1\|\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'^*))\|_2$. Then we do a calculation,

$$\begin{aligned}
\mathbb{E}L^2_{\sigma_{\mathcal{K}}} &\leqslant \frac{1}{r}\|\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'^*))\|_2 \int_0^\infty dw_1 \left\{ \int_{-\infty}^\infty dw_2 ... \int_{-\infty}^\infty dw_d (w_1 \sum_{j=1}^d w_j^2) p(w_2, ..., w_d) \right\} p(w_1) \\
&= \frac{1}{r}\|\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'^*))\|_2 \int_0^\infty dw_1 \{w_1^3 + (d-1)w_1) p(w_1)\} \\
&= \frac{1}{r}\|\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'^*))\|_2 \int_0^\infty dw_1 \{(d+3)w_1 p(w_1)\} \\
&= \frac{1}{r}\|\sigma(\mathbf{f}^{\ell-1}(\mathbf{x}'^*))\|_2 \frac{\sqrt{2}(d+3)}{\sqrt{\pi}} \\
&\leqslant \frac{\sqrt{2}(d+3)c_{\mathcal{M}}}{\sqrt{\pi}r}
\end{aligned}$$

Therefore, by Markov's inequality,

$$\mathbb{P}(L_{\sigma_{\mathcal{K}}} \geqslant \frac{\epsilon}{4\delta}) \leqslant \mathbb{E}L^2_{\sigma_{\mathcal{K}}} \frac{16\delta^2}{\epsilon^2} \leqslant \frac{16\sqrt{2}(d+3)c_{\mathcal{M}}\delta^2}{\sqrt{\pi}\epsilon^2 r}.$$

Then using Eq. (A.113), with probability at least $1 - \frac{16\sqrt{2}(d+3)c_{\mathcal{M}}\delta^2}{\sqrt{\pi}\epsilon^2 r}$,

$$|u(\mathbf{x}, \mathbf{x}') - u(\mathbf{x}_0, \mathbf{x}_0')| \leqslant \frac{\epsilon}{2}.$$

This inequality combined with Eq. (A.112) enables us to conclude that

$$\sup_{\sigma(\mathbf{f}^{\ell-1}(\mathbf{x})), \sigma(\mathbf{f}^{\ell-1}(\mathbf{x}')) \in \mathcal{M}} |\hat{\mathcal{K}}^\ell(\mathbf{f}^{\ell-1}(\mathbf{x}), \mathbf{f}^{\ell-1}(\mathbf{x}')) - \mathcal{K}^\ell(\mathbf{f}^{\ell-1}(\mathbf{x}), \mathbf{f}^{\ell-1}(\mathbf{x}'))| \leqslant \epsilon.$$

with probability at least $1 - \frac{16\sqrt{2}(d+3)c_{\mathcal{M}}\delta^2}{\sqrt{\pi}\epsilon^2 r} - 2\exp\left(\log(T^2) - \frac{r\epsilon^2}{8\nu^2_{\mathcal{M}}}\right)$. Recall that $T = (4diam(\mathcal{M})/\delta)^d$, so the probability has a format of $1 - \kappa_1\delta^2 - \kappa_2\delta^{-2d}$ for δ. By setting $\delta = \frac{\kappa_2}{\kappa_1}^{\frac{1}{2+2d}}$, we have the probability as $1 - 2\kappa_1^{\frac{2d}{2+2d}}\kappa_2^{\frac{2}{2+2d}}$. So the probability is at least

$$1 - 2^{10}\left(\frac{c_{\mathcal{M}}(d+3)diam(\mathcal{M})^2}{\epsilon^2 r}\right)^{\frac{d}{1+d}} \exp\left\{-\frac{r\epsilon^2}{8(1+d)v_{\mathcal{M}}^2}\right\}.$$

□

The relation between random feature Cutajar et al. (2017) and inducing points approximation Salimbeni and Deisenroth (2017)

First, we review the algorithm in Salimbeni and Deisenroth (2017) based on inducing points and doubly stochastic variational inference.

In the background section, we introduced that a L layer DGP can be represented by

$$p(\mathbf{y}, \{\mathbf{F}^\ell\}_{\ell=1}^L) = \prod_{i=1}^n p(y_i|f_i^L) \prod_{\ell=1}^L p(\mathbf{F}^\ell|\mathbf{F}^{\ell-1}),$$

where $\mathbf{F}^\ell \in \mathbb{R}^{n\times d}$ for $0 \leqslant \ell < L$ and $\mathbf{F}^L \in \mathbb{R}^{n\times 1}$, with $\mathbf{F}^\ell|\mathbf{F}^{\ell-1} \sim N(0, \mathcal{K}^\ell(\mathbf{F}^{\ell-1}, \mathbf{F}^{\ell-1}))$.

In Salimbeni and Deisenroth (2017), they further define an additional set of m inducing points $\mathbf{Z}^\ell = (\mathbf{z}_1^\ell, ..., \mathbf{z}_m^\ell)$ for each layer $0 \leqslant \ell < L$. We use the notation $\mathbf{u}^\ell = f^\ell(\mathbf{Z}^{\ell-1})$ for the function values at the inducing points. Since we have d output on layer ℓ, we use $\mathbf{U}^\ell \in \mathbb{R}^{m\times d}$ for the function value matrix at the inducing points. By the definition of GP, the joint density $p(\mathbf{F}^\ell, \mathbf{U}^\ell)$ is a Gaussian distribution given inputs from previous layer. Therefore, we have the joint posterior of $\mathbf{y}, \{\mathbf{F}^\ell, \mathbf{U}^\ell\}_{\ell=1}^L$ is

$$p(\mathbf{y}, \{\mathbf{F}^\ell, \mathbf{U}^\ell\}_{\ell=1}^L) = \prod_{i=1}^n p(y_i|f_i^L) \prod_{\ell=1}^L p(\mathbf{F}^\ell|\mathbf{U}^\ell; \mathbf{F}^{\ell-1}, \mathbf{Z}^{\ell-1}) p(\mathbf{U}^\ell; \mathbf{Z}^{\ell-1}).$$

The posterior of $\{\mathbf{F}^\ell, \mathbf{U}^\ell\}_{\ell=1}^L$ is intractable, so the authors in Salimbeni and Deisenroth (2017) define the variational posterior

$$q(\{\mathbf{F}^\ell, \mathbf{U}^\ell\}_{\ell=1}^L) = \prod_{\ell=1}^L p(\mathbf{F}^\ell|\mathbf{U}^\ell; \mathbf{F}^{\ell-1}, \mathbf{Z}^{\ell-1}) q(\mathbf{U}^\ell),$$

with $q(\mathbf{U}^\ell) = \prod_{j=1}^{d} q(\mathbf{U}_{\cdot j}^\ell)$ and $q(\mathbf{U}_{\cdot j}^\ell) \sim N(\mathbf{m}_j^\ell, \mathbf{S}_j^\ell)$. Then they calculate the evidence lower bound of the DGP, which is

$$\mathbf{ELBO}_{DGP} = \mathbb{E}_{q(\{\mathbf{F}^\ell, \mathbf{U}^\ell\}_{\ell=1}^L)} \left[\frac{p(\mathbf{y}, \{\mathbf{F}^\ell, \mathbf{U}^\ell\}_{\ell=1}^L}{q(\{\mathbf{F}^\ell, \mathbf{U}^\ell\}_{\ell=1}^L)} \right].$$

Based on the definition of $q(\{\mathbf{F}^\ell, \mathbf{U}^\ell\}_{\ell=1}^L)$, we can simplify $\mathbf{ELBO}_{DGP}$ and show that it is equal as

$$\mathbf{ELBO}_{DGP} = \sum_{i=1}^{n} \mathbb{E}_{q(f_i^L)}[\log p(y_i | f_i^L)] - \sum_{\ell=1}^{L} KL[q(\mathbf{U}^\ell) | p(\mathbf{U}^\ell; \mathbf{Z}^{\ell-1})]. \tag{A.114}$$

From Salimbeni and Deisenroth (2017), after marginalizing the inducing variables from each layer analytically, we can show that

$$q(\{\mathbf{F}^\ell\}_{\ell=1}^L) = \prod_{\ell=1}^{L} q(\mathbf{F}^\ell | \mathbf{m}^\ell, \mathbf{S}^\ell; \mathbf{F}^{\ell-1}, \mathbf{Z}^{\ell-1}) = \prod_{\ell=1}^{L} N(\mathbf{F}^\ell | \boldsymbol{\mu}^\ell, \boldsymbol{\Sigma}^\ell) = \prod_{\ell=1}^{L} \prod_{j=1}^{d} N(\mathbf{F}_{\cdot j}^\ell | \tilde{\boldsymbol{\mu}}_j^\ell, \tilde{\boldsymbol{\Sigma}}_j^\ell). \tag{A.115}$$

Here,

$$\tilde{\boldsymbol{\mu}}_j^\ell = \mathbf{ff}(\mathbf{F}^{\ell-1})^\top \mathbf{m}_{\cdot j}^\ell$$

$$\tilde{\boldsymbol{\Sigma}}_j^\ell = \mathcal{K}^\ell(\mathbf{F}^{\ell-1}, \mathbf{F}^{\ell-1}) - \mathbf{ff}(\mathbf{F}^{\ell-1})^\top (\mathcal{K}^\ell(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1}) - \mathbf{S}_{\cdot j}) \mathbf{ff}(\mathbf{F}^{\ell-1})$$

with $\mathbf{ff}(\mathbf{F}^{\ell-1}) = \mathcal{K}^\ell(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1})^{-1} \mathcal{K}^\ell(\mathbf{Z}^{\ell-1}, \mathbf{F}^{\ell-1})$.

From Eq. (A.114), we only need to get $q(f_i^L)$ from $q(\{\mathbf{F}^\ell\}_{\ell=1}^L)$ for sample i with $1 \leqslant i \leqslant n$. In Salimbeni and Deisenroth (2017), they point out that based on the format of Eq. (A.115),

$$q(f_i^L) = \int \cdots \int \prod_{\ell=1}^{L} q(\mathbf{F}_{i\cdot}^\ell | \mathbf{m}_{i\cdot}^\ell, \mathbf{S}_{i,}^\ell; \mathbf{F}_{i\cdot}^{\ell-1}, \mathbf{Z}^{\ell-1}) d\mathbf{F}_{i\cdot}^{\ell-1},$$

which means that the ith marginal of the final layer of the variational DGP for sample i depends only on the ith marginals of all the other layers.

Proof of Theorem 2

Theorem A.30. *Using the variational approximation Salimbeni and Deisenroth (2017) for the posterior of a DGP defined on $\{\hat{\mathcal{K}}^\ell\}_{\ell=1}^L$ with inducing points, we obtain exactly the same variational posterior $q(\{\mathbf{F}^\ell\}_{\ell=1}^L)$ and evidence lower bound **ELBO** as the variational posterior for $\mathcal{N}(\mathcal{S})$.*

Proof. To show the equivalence of evidence lower bound, we only need to guarantee that $q(\mathbf{F}_{i.}^\ell|\mathbf{m}_{i.}^\ell, \mathbf{S}_{i,}^\ell; \mathbf{F}_{i.}^{\ell-1}, \mathbf{Z}^{\ell-1})$ and $KL[q(\mathbf{U}^\ell)|p(\mathbf{U}^\ell; \mathbf{Z}^{\ell-1})]$ are the same as the relevant values for $\mathcal{N}(\mathcal{S})$ for all $1 \leqslant i \leqslant n$ and $1 \leqslant \ell \leqslant L$. We also need to show the equivalence between variational posterior $q(\{\mathbf{F}^\ell\}_{\ell=1}^L)$ for the two methods. All those can be satisfied by showing the equivalence that $q(\mathbf{F}^\ell|\mathbf{m}^\ell, \mathbf{S}^\ell; \mathbf{F}^{\ell-1}, \mathbf{Z}^{\ell-1})$ and $KL[q(\mathbf{U}^\ell)|p(\mathbf{U}^\ell; \mathbf{Z}^{\ell-1})]$ are the same as the relevant values for $\mathcal{N}(\mathcal{S})$ for all $1 \leqslant \ell \leqslant L$. For both two methods, since for each layer ℓ, the d outputs are independent, so the posterior distribution can be decomposed into a product of d terms and the KL divergence can be decomposed into a summation of d terms. We only need to prove the result for a single j with $1 \leqslant j \leqslant d$ and a single ℓ with $1 \leqslant \ell \leqslant L$.

Based on Eq. (A.115), for $q(\mathbf{F}_{.j}^\ell|\mathbf{m}_{.j}^\ell, \mathbf{S}_{.j}^\ell; \mathbf{F}^{\ell-1}, \mathbf{Z}^{\ell-1})$, the mean and variances are

$$\tilde{\boldsymbol{\mu}}_j^\ell = \mathbf{ff}(\mathbf{F}^{\ell-1})^\top \mathbf{m}_{.j}^\ell$$

$$\tilde{\boldsymbol{\Sigma}}_j^\ell = \hat{\mathcal{K}}^\ell(\mathbf{F}^{\ell-1}, \mathbf{F}^{\ell-1}) - \mathbf{ff}(\mathbf{F}^{\ell-1})^\top(\hat{\mathcal{K}}^\ell(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1}) - \mathbf{S}_{.j})\mathbf{ff}(\mathbf{F}^{\ell-1})$$

with $\mathbf{ff}(\mathbf{F}^{\ell-1}) = \hat{\mathcal{K}}^\ell(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1})^{-1}\hat{\mathcal{K}}^\ell(\mathbf{Z}^{\ell-1}, \mathbf{F}^{\ell-1})$. We can decompose the kernel into $\hat{\mathcal{K}}^\ell(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1}) = \Phi^\ell(\mathbf{Z}^{\ell-1})\Phi^\ell(\mathbf{Z}^{\ell-1})^\top$. Now we choose the number of inducing points m as $m = r$, then we have a square matrix $\Phi^\ell(\mathbf{Z}^{\ell-1})$ with each entry is independently identically from a distribution based on the random feature weight vector $\mathbf{w}_j$ and the random inducing points $\mathbf{Z}_{i.}^{\ell-1}$.

For continuous $\sigma_{\mathcal{K}}$ and σ, every entry in $\Phi^{\ell}(\mathbf{Z}^{\ell-1})$ is absolutely continuous with respect to Lebesgue measure since we can define density function. Then based on random matrix theory Rudelson (2008); Tao (2012), the square matrix $\Phi^{\ell}(\mathbf{Z}^{\ell-1})$ is almost surely invertible. Therefore, we treat $\Phi^{\ell}(\mathbf{Z}^{\ell-1})$ as an invertible matrix in following analysis.

Replace $\hat{\mathcal{K}}^{\ell}(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1})$ by $\Phi^{\ell}(\mathbf{Z}^{\ell-1})\Phi^{\ell}(\mathbf{Z}^{\ell-1})^{\mathsf{T}}$, we have that

$$\tilde{\boldsymbol{\mu}}_{j}^{\ell} = \Phi^{\ell}(\mathbf{F}^{\ell-1})(\Phi^{\ell}(\mathbf{Z}^{\ell-1})^{\mathsf{T}}\Phi^{\ell}(\mathbf{Z}^{\ell-1}))^{-1}\Phi^{\ell}(\mathbf{Z}^{\ell-1})^{\mathsf{T}}\mathbf{m}_{\cdot j}^{\ell}$$

$$\tilde{\boldsymbol{\Sigma}}_{j}^{\ell} = \Phi^{\ell}(\mathbf{F}^{\ell-1})\Phi^{\ell}(\mathbf{F}^{\ell-1})^{\mathsf{T}} - \mathbf{ff}(\mathbf{F}^{\ell-1})^{\mathsf{T}}(\Phi^{\ell}(\mathbf{Z}^{\ell-1})\Phi^{\ell}(\mathbf{Z}^{\ell-1})^{\mathsf{T}} - \mathbf{S}_{\cdot j})\mathbf{ff}(\mathbf{F}^{\ell-1})$$

Through simple algebra, we get that

$$\begin{aligned} \tilde{\boldsymbol{\mu}}_{j}^{\ell} &= \mathbf{ff}(\mathbf{F}^{\ell-1})^{\mathsf{T}}\mathbf{m}_{\cdot j}^{\ell} \\ \tilde{\boldsymbol{\Sigma}}_{j}^{\ell} &= \mathbf{ff}(\mathbf{F}^{\ell-1})^{\mathsf{T}}\mathbf{S}_{\cdot j}\mathbf{ff}(\mathbf{F}^{\ell-1}) \end{aligned} \tag{A.116}$$

Since $\Phi^{\ell}(\mathbf{Z}^{\ell-1})$ is invertible, we define

$$\mathbf{m}_{\cdot j}^{\ell} = \Phi^{\ell}(\mathbf{Z}^{\ell-1})\boldsymbol{\mu}_{j,new}^{\ell}$$

$$\mathbf{S}_{\cdot j} = \Phi^{\ell}(\mathbf{Z}^{\ell-1})\boldsymbol{\Sigma}_{j,new}^{\ell}\Phi^{\ell}(\mathbf{Z}^{\ell-1})^{\mathsf{T}},$$

and plug them into Eq. (A.116) then we get

$$\begin{aligned} \tilde{\boldsymbol{\mu}}_{j}^{\ell} &= \Phi^{\ell}(\mathbf{F}^{\ell-1})\boldsymbol{\mu}_{j,new}^{\ell} \\ \tilde{\boldsymbol{\Sigma}}_{j}^{\ell} &= \Phi^{\ell}(\mathbf{F}^{\ell-1})\boldsymbol{\Sigma}_{j,new}^{\ell}\Phi^{\ell}(\mathbf{F}^{\ell-1})^{\mathsf{T}}. \end{aligned} \tag{A.117}$$

In our BNN construction for $\mathcal{N}(\mathcal{S})$, the variational posterior over $\mathbf{V}$ leads to $\mathbf{F}_{\cdot j}^{\ell} = \Phi^{\ell}(\mathbf{F}^{\ell-1})\mathbf{v}_{j}^{\ell}$ with $\mathbf{v}_{j}^{\ell} \sim \mathrm{N}(\boldsymbol{\mu}_{j,new}^{\ell}, \boldsymbol{\Sigma}_{j,new}^{\ell})$. Then we have that

$$\mathbf{F}_{\cdot j}^{\ell} \sim \mathrm{N}(\Phi^{\ell}(\mathbf{F}^{\ell-1})\boldsymbol{\mu}_{j,new}^{\ell}, \Phi^{\ell}(\mathbf{F}^{\ell-1})\boldsymbol{\Sigma}_{j,new}^{\ell}\Phi^{\ell}(\mathbf{F}^{\ell-1})^{\mathsf{T}}).$$

That is identical with the results from inducing points method in Eq. (A.117). Based on this construction, we also have that

$$\mathbf{U}_{\cdot j}^{\ell} = \Phi^{\ell}(\mathbf{Z}^{\ell-1})\mathbf{v}_j^{\ell}$$

Since the KL divergence is invariant under parameter transformations, we have that

$$\mathrm{KL}(q(\mathbf{U}_{\cdot j}^{\ell})||p(\mathbf{U}_{\cdot j}^{\ell})) = \mathrm{KL}(q(\mathbf{v}_j^{\ell})||p(\mathbf{v}_j^{\ell}))$$

□

Proof of Theorem 3

For a kernel $\mathcal{K}^{\ell}$ belongs to a general class, we can still use a similar technique as in the proof of Theorem 2 to show the equivalence. However, this time we cannot use the random feature matrix $\Phi^{\ell}(\mathbf{F}^{\ell-1})\Phi^{\ell}(\mathbf{F}^{\ell-1})^{\mathsf{T}}$ to approximate $\mathcal{K}^{\ell}(\mathbf{F}^{\ell-1}, \mathbf{F}^{\ell-1})$. It turns out that a good replacement for $\Phi^{\ell}(\mathbf{F}^{\ell-1})$ to approximate the basis of $\mathcal{K}^{\ell}(\mathbf{F}^{\ell-1}, \mathbf{F}^{\ell-1})$ is $\mathcal{K}^{\ell}(\mathbf{F}^{\ell-1}, \mathbf{Z}^{\ell-1})\mathcal{K}^{\ell}(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1})^{-1/2}$ which we will show shortly. The proof technique for Theorem 3 is similar as the technique for Theorem 2. However, for a general class of $\mathcal{K}^{\ell}$, $\mathcal{K}^{\ell}(\mathbf{F}^{\ell-1}, \mathbf{F}^{\ell-1})$ can be full rank which is equal to sample size n. Therefore, the difference from the approximation using the rank r basis $\mathcal{K}^{\ell}(\mathbf{F}^{\ell-1}, \mathbf{Z}^{\ell-1})\mathcal{K}^{\ell}(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1})^{-1/2}$ is

$$\mathcal{K}^{\ell}(\mathbf{F}^{\ell-1}, \mathbf{F}^{\ell-1}) - \mathcal{K}^{\ell}(\mathbf{F}^{\ell-1}, \mathbf{Z}^{\ell-1})\mathcal{K}^{\ell}(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1})^{-1}\mathcal{K}^{\ell}(\mathbf{Z}^{\ell-1}, \mathbf{F}^{\ell-1}).$$

This is the constant offset that does not depend on training which we mention in Theorem 3. For the optimization of **ELBO**, only the diagonal terms in this offset matrix is used so we can also add this into BNN as a bias term with random weight that we do not train.

Remark. After the optimization of **ELBO**, one can get the uncertainty estimates from the variational posterior. If

$$\mathcal{K}^{\ell}(\mathbf{F}^{\ell-1}, \mathbf{F}^{\ell-1}) - \mathcal{K}^{\ell}(\mathbf{F}^{\ell-1}, \mathbf{Z}^{\ell-1})\mathcal{K}^{\ell}(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1})^{-1}\mathcal{K}^{\ell}(\mathbf{Z}^{\ell-1}, \mathbf{F}^{\ell-1})$$

is not present, then one can choose $\mathbf{V}$ from its variational posterior and the output estimates for every samples directly come from one pass of feed-forward neural network. However, when $\mathcal{K}^{\ell}(\mathbf{F}^{\ell-1}, \mathbf{F}^{\ell-1}) - \mathcal{K}^{\ell}(\mathbf{F}^{\ell-1}, \mathbf{Z}^{\ell-1})\mathcal{K}^{\ell}(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1})^{-1}\mathcal{K}^{\ell}(\mathbf{Z}^{\ell-1}, \mathbf{F}^{\ell-1})$ exists, n passes of feed-forward neural network computation for n samples need to depend on each other to derive the outputs. In Salimbeni and Deisenroth (2017), they also permit the prior of DGP to have a non-zero mean function, which can lead to another offset if the prior mean of DGP at each layer is non-zero. Similarly, this offset does not depend on training and can be included into BNN as a bias term with random weights that we do not train. Another term that is usually discussed in DGP is the noisy corruption. In this result for general kernel $\mathcal{K}$ in Thoerem 3, in Salimbeni and Deisenroth (2017), the authors show that the noisy corruption can be included into the kernel $\mathcal{K}$. For our earlier result in Theorem 2, we do not include the noisy corruption in intermediate layers, since the complexity of intermediate function is already restricted by the rank r. It does not overfit the data so we do not need the noisy corruption which is usually used to avoid overfitting when the kernel basis has infinite dimension which can be super expressive.

Now we review the definition of IPB and Theorem 3, then we present the proof for Theorem 3.

Definition A.31. *For a kernel $\mathcal{K}$,* IPB *can be constructed by choosing r additional points $\mathbf{Z}$ (inducing points), taking the inputs $\mathbf{x}$ and outputting an r-dimension vector $\mathcal{K}(\mathbf{x}, \mathbf{Z})\mathcal{K}(\mathbf{Z}, \mathbf{Z})^{-1/2}$.*

Theorem A.32. *Using the variational approximation Salimbeni and Deisenroth (2017) for the posterior of a DGP defined on $\{\mathcal{K}^{\ell}\}_{\ell=1}^{L}$ with inducing points, we can obtain the same variational posterior $q(\{\mathbf{F}^{\ell}\}_{\ell=1}^{L})$ and evidence lower bound ELBO as the variational posterior for $\mathcal{N}(\mathcal{S})$ (with* IPB*) except a constant offset that does not depend on training.*

Proof. To show the equivalence of evidence lower bound, we only need to guarantee that $q(\mathbf{F}_{i,}^{\ell}|\mathbf{m}_{i,}^{\ell}, \mathbf{S}_{i,}^{\ell}; \mathbf{F}_{i,}^{\ell-1}, \mathbf{Z}^{\ell-1})$ and $KL[q(\mathbf{U}^{\ell})|p(\mathbf{U}^{\ell}; \mathbf{Z}^{\ell-1})]$ are the same as the relevant values for $\mathcal{N}(\mathcal{S})$ for all $1 \leqslant i \leqslant n$ and $1 \leqslant \ell \leqslant L$. We also need to show the equivalence between variational posterior $q(\{\mathbf{F}^{\ell}\}_{\ell=1}^{L})$ for the two methods. All those can be satisfied by showing the equivalence that $q(\mathbf{F}^{\ell}|\mathbf{m}^{\ell}, \mathbf{S}^{\ell}; \mathbf{F}^{\ell-1}, \mathbf{Z}^{\ell-1})$ and

$\mathrm{KL}[q(\mathbf{U}^\ell)|p(\mathbf{U}^\ell;\mathbf{Z}^{\ell-1})]$ are the same as the relevant values for $\mathcal{N}(\mathcal{S})$ for all $1 \leqslant \ell \leqslant L$. For both two methods, since for each layer ℓ, the d outputs are independent, so the posterior distribution can be decomposed into a product of d terms and the KL divergence can be decomposed into a summation of d terms. We only need to prove the result for a single j with $1 \leqslant j \leqslant d$ and a single ℓ with $1 \leqslant \ell \leqslant L$.

Based on Eq. (A.115), for $q(\mathbf{F}_{:j}^\ell|\mathbf{m}_{:j}^\ell, \mathbf{S}_{:j}^\ell; \mathbf{F}^{\ell-1}, \mathbf{Z}^{\ell-1})$, the mean and variances are

$$\tilde{\boldsymbol{\mu}}_j^\ell = \mathbf{ff}(\mathbf{F}^{\ell-1})^\top \mathbf{m}_{:j}^\ell$$

$$\tilde{\boldsymbol{\Sigma}}_j^\ell = \mathcal{K}^\ell(\mathbf{F}^{\ell-1}, \mathbf{F}^{\ell-1}) - \mathbf{ff}(\mathbf{F}^{\ell-1})^\top(\mathcal{K}^\ell(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1}) - \mathbf{S}_{:j})\mathbf{ff}(\mathbf{F}^{\ell-1})$$

with $\mathbf{ff}(\mathbf{F}^{\ell-1}) = \mathcal{K}^\ell(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1})^{-1}\mathcal{K}^\ell(\mathbf{Z}^{\ell-1}, \mathbf{F}^{\ell-1})$.

We use r to refer the number of inducing points and **we denote the** IPB **block matrix** $\mathcal{K}^\ell(\mathbf{F}^{\ell-1}, \mathbf{Z}^{\ell-1})\mathcal{K}^\ell(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1})^{-1/2}$ as $\mathbf{IP}^\ell(\mathbf{F}^{\ell-1})$, then we notice that

$$\begin{aligned}
\tilde{\boldsymbol{\mu}}_j^\ell &= \mathbf{IP}^\ell(\mathbf{F}^{\ell-1})\mathcal{K}^\ell(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1})^{-1/2}\mathbf{m}_{:j}^\ell \\
\tilde{\boldsymbol{\Sigma}}_j^\ell &= \mathbf{IP}^\ell(\mathbf{F}^{\ell-1})\mathcal{K}^\ell(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1})^{-1/2}\mathbf{S}_{:j}\mathcal{K}^\ell(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1})^{-1/2}\mathbf{IP}^\ell(\mathbf{F}^{\ell-1})^\top \\
&\quad + \mathcal{K}^\ell(\mathbf{F}^{\ell-1}, \mathbf{F}^{\ell-1}) - \mathcal{K}^\ell(\mathbf{F}^{\ell-1}, \mathbf{Z}^{\ell-1})\mathcal{K}^\ell(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1})^{-1}\mathcal{K}^\ell(\mathbf{Z}^{\ell-1}, \mathbf{F}^{\ell-1}).
\end{aligned} \tag{A.118}$$

We have discussed the second term in $\tilde{\boldsymbol{\Sigma}}_j^\ell$ which is a constant offset that does not depend on training. Therefore we assume it to be zero in following analysis then we get the exactly same result as our variational posterior approximation for $\mathcal{N}(\mathcal{S})$ when IPB is used.

We define

$$\mathbf{m}_{:j}^\ell = \mathcal{K}^\ell(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1})^{1/2}\boldsymbol{\mu}_{j,new}^\ell$$

$$\mathbf{S}_{:j} = \mathcal{K}^\ell(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1})^{1/2}\boldsymbol{\Sigma}_{j,new}^\ell\mathcal{K}^\ell(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1})^{1/2},$$

and plug them into Eq. (A.118) then we get

$$\begin{aligned}\tilde{\mu}_j^\ell &= \mathbf{IP}^\ell(\mathbf{F}^{\ell-1})\mu_{j,new}^\ell \\ \tilde{\Sigma}_j^\ell &= \mathbf{IP}^\ell(\mathbf{F}^{\ell-1})\Sigma_{j,new}^\ell \mathbf{IP}^\ell(\mathbf{F}^{\ell-1})^\top.\end{aligned} \tag{A.119}$$

In our BNN construction for $\mathcal{N}(\mathcal{S})$, the variational posterior over $\mathbf{V}$ leads to $\mathbf{F}_{\cdot j}^\ell = \mathbf{IP}^\ell(\mathbf{F}^{\ell-1})\mathbf{v}_j^\ell$ with $\mathbf{v}_j^\ell \sim N(\mu_{j,new}^\ell, \Sigma_{j,new}^\ell)$. Then we have that

$$\mathbf{F}_{\cdot j}^\ell \sim N(\mathbf{IP}^\ell(\mathbf{F}^{\ell-1})\mu_{j,new}^\ell, \mathbf{IP}^\ell(\mathbf{F}^{\ell-1})\Sigma_{j,new}^\ell \mathbf{IP}^\ell(\mathbf{F}^{\ell-1})^\top).$$

That is identical with the results from inducing points method in Eq. (A.119). Based on this construction, we also have that

$$\mathbf{U}_{\cdot j}^\ell = \mathbf{IP}^\ell(\mathbf{Z}^{\ell-1})\mathbf{v}_j^\ell = \mathcal{K}^\ell(\mathbf{Z}^{\ell-1}, \mathbf{Z}^{\ell-1})^{1/2}\mathbf{v}_j^\ell$$

Since the KL divergence is invariant under parameter transformations, we have that

$$KL(q(\mathbf{U}_{\cdot j}^\ell)||p(\mathbf{U}_{\cdot j}^\ell)) = KL(q(\mathbf{v}_j^\ell)||p(\mathbf{v}_j^\ell))$$

□

Proof of Theorem 4

The post-training ANOVA decomposition is

$$\mathbf{I}_T^n(\mathbf{x}_T) = \prod_{i\in T}(I_{x_i} - \mathbb{E}_{x_i}^n)\prod_{j\notin T}\mathbb{E}_{x_j}^n f(x_1, ..., x_p), \tag{A.120}$$

Theorem A.33. *If there exist inputs clusters* $\{T_j^*\}_{j=1}^{k^*}$ *such that* $f^*(\mathbf{x}) = \sum_{j=1}^{k^*} g_j^*(\mathbf{x}_{T_j^*})$ *with* k^* *at the order of polynomial in* p *and* $c = \max_{j=1}^{k^*}|T_j^*| = O(\log p)$*, then there exists a trained AddNN that predicts* $\mathbf{y}$ *well and restricts the number of possible interactions at polynomial in* p*. Further, if every sub neural network has* L *layers with* d *hidden units, then the computation complexity of measure* (A.120) *is at most* $n^c k^* d^{2L-1}$*, which is also polynomial in* p*.*

Proof. In that case, there exists a trained AddNN f which is $\hat{f}(\mathbf{x}) = \sum_{j=1}^{k^*} \hat{g}_j(\mathbf{x}_{T_j^*})$ with the same k^* and inputs clusters $\{T_j^*\}_{j=1}^{k^*}$. For such $\hat{f}$, without knowing the truth, every possible interaction is a subset of one inputs cluster T_j^* for some j. Therefore, the number of possible interactions is bounded by $\sum_{j=1}^{k^*} 2^{|T_j^*|} \leqslant k^* 2^c$, so it is polynomial in p. For an interaction among a subset S, we have that

$$\begin{aligned}
\mathbf{I}_S^n(\mathbf{x}_S) &= \prod_{i\in S}(I_{x_i} - \mathbb{E}_{x_i}^n) \prod_{j\notin S} \mathbb{E}_{x_j}^n \hat{f}(x_1, ..., x_p) \\
&= \prod_{i\in S}(I_{x_i} - \mathbb{E}_{x_i}^n) \prod_{j\notin S} \mathbb{E}_{x_j}^n \sum_{m=1}^{k^*} \hat{g}_m(\mathbf{x}_{T_m^*}) \\
&= \sum_{m=1}^{k^*} \prod_{i\in S}(I_{x_i} - \mathbb{E}_{x_i}^n) \prod_{j\notin S} \mathbb{E}_{x_j}^n \hat{g}_m(\mathbf{x}_{T_m^*}) \\
&= \sum_{m:S\subseteq T_m^*} \prod_{i\in S}(I_{x_i} - \mathbb{E}_{x_i}^n) \prod_{j\notin S} \mathbb{E}_{x_j}^n \hat{g}_m(\mathbf{x}_{T_m^*}),
\end{aligned}$$

because for m that $S \not\subseteq T_m^*$, then there exists an $i_0 \in S$ such that $i_0 \notin T_m^*$, then

$$\prod_{i\in S}(I_{x_i} - \mathbb{E}_{x_i}^n) \prod_{j\notin S} \mathbb{E}_{x_j}^n \hat{g}_m(\mathbf{x}_{T_m^*}) = 0$$

because $(I_{x_{i_0}} - \mathbb{E}_{x_{i_0}}^n)\hat{g}_m(\mathbf{x}_{T_m^*}) = \hat{g}_m(\mathbf{x}_{T_m^*}) - \hat{g}_m(\mathbf{x}_{T_m^*}) = 0$. We can further simply $\mathbf{I}_S^n(\mathbf{x}_S)$,

$$\begin{aligned}
\mathbf{I}_S^n(\mathbf{x}_S) &= \sum_{m:S\subseteq T_m^*} \prod_{i\in S}(I_{x_i} - \mathbb{E}_{x_i}^n) \prod_{j\notin S} \mathbb{E}_{x_j}^n \hat{g}_m(\mathbf{x}_{T_m^*}) \\
&= \sum_{m:S\subseteq T_m^*} \prod_{i\in S}(I_{x_i} - \mathbb{E}_{x_i}^n) \prod_{j\notin S, j\in T_m^*} \mathbb{E}_{x_j}^n \hat{g}_m(\mathbf{x}_{T_m^*}).
\end{aligned}$$

Therefore, the computation complexity of $\mathbf{I}_S^n(\mathbf{x}_S)$ is equal to the computation complexity for

$$\cup_{m:S\subseteq T_m^*}\left\{\prod_{i\in S}(I_{x_i}-\mathbb{E}_{x_i}^n)\prod_{j\notin S,j\in T_m^*}\mathbb{E}_{x_j}^n\hat{g}_m(\mathbf{x}_{T_m^*})\right\}, \tag{A.121}$$

which is the union of $|m : S \subseteq T_m^*|$ functions where each function involves the calculation of feed-forward neural network and empirical evaluation. Therefore, to compute all interactions, we need to compute the union of (A.121) for all possible interactions S. Because every possible interaction is a subset of one inputs cluster T_j^* for some j, we can exchange the order of unions and we get that the computation complexity for all interactions is equal as evaluating

$$\cup_m \cup_{S\subseteq T_m^*}\left\{\prod_{i\in S}(I_{x_i}-\mathbb{E}_{x_i}^n)\prod_{j\notin S,j\in T_m^*}\mathbb{E}_{x_j}^n\hat{g}_m(\mathbf{x}_{T_m^*})\right\}. \tag{A.122}$$

To compute $\cup_{S\subseteq T_m^*}\left\{\prod_{i\in S}(I_{x_i}-\mathbb{E}_{x_i}^n)\prod_{j\notin S,j\in T_m^*}\mathbb{E}_{x_j}^n\hat{g}_m\left(\mathbf{x}_{T_m^*}\right)\right\}$ for some m, we only need to compute

$$M_{T_m^*}^n(\mathbf{x}_{T_m^*})=\prod_{i\in T_m^*}I_{x_i}\hat{g}\left(\mathbf{x}_{T_m^*}\right),$$

which involves all the evaluations of the feed-forward neural network. To evaluate $\prod_{i\in S}(I_{x_i}-\mathbb{E}_{x_i}^n)\prod_{j\notin S,j\in T_m^*}\mathbb{E}_{x_j}^n\hat{g}_m\left(\mathbf{x}_{T_m^*}\right)$ for $S\subseteq T_m^*$, the computation only involves basic addition operation given $M_{T_m^*}^n$. Therefore, we show that the evaluation of all possible interactions has the same computation complexity as evaluating

$$\cup_m\{M_{T_m^*}^n(\mathbf{x}_{T_m^*})\}=\cup_m\{\prod_{i\in T_m^*}I_{x_i}\hat{g}\left(\mathbf{x}_{T_m^*}\right)\}. \tag{A.123}$$

For every member in the union regarding m, the evaluation complexity is $n^{|T_m^*|}d^{2L-1}\leqslant n^c d^{2L-1}$. Therefore, the computation complexity regarding (A.120) for all possible interactions based on model $\hat{f}$ is bounded by $n^c k^* d^{2L-1}$, which is polynomial in p when k^* is polynomial in p and c is the order of $O(\log p)$. □

Remark 1. When the truth function f^* is additive, the function class of AddNN includes a member that can compute interactions and outputs from the model efficiently while the function class of an arbitrary NN make it impossible to learn such a model and always involve computation that is exponential in p. In practice, we always use group Lasso type penalty on the first layer to encourage each sub neural network to depend on few inputs to approach the truth function f^*.

Remark 2. We can use the measure in Zeiler and Fergus (2014) as well, which can be seen as choosing one sample baseline instead of the average baseline in (A.120). In other words, now we use operation $\delta_{x_i}(\mathbf{x}_i^0)$ to replace $\mathbb{E}_{x_i}^n$ in (A.120) based on one sample $\mathbf{x}^0$ for $1 \leqslant i \leqslant n$. Then the computation complexity of measure (A.120) does not depend on n both for AddNN and NN. In that case, the number of possible interactions for AddNN is still at polynomial in p and the number is exponential in p for an arbitrary NN. Also, the computation complexity of measure (A.120) is $k^* d^{2L-1}$ for AddNN and $(k^* d)^{2L-1}$ for an arbitrary NN with L layers and $k^* d$ hidden units where the NN requires $k^{*2(L-1)}$ times more computation. The choice of $\mathbb{E}^n$ as baseline, compared to $\delta(\mathbf{x}^0)$, is better for comparison with the population and can lead to a useful measure $||I_T^n(\mathbf{x}_T)||_{2,n}$ (the empirical ℓ_2 norm of the interaction), which can be used to detect the interactions. We give an example of their difference for explanation in the decision making process. For one who is interested in making an investment with x_1 dollars, the $\mathbb{E}^n$ baseline informs that, based on this investment, how much more one can earn than the average of the money that people earn. On the other side, the $\delta(\mathbf{x}_0)$ baseline informs that, at the current investment with x_0 dollars, if all other factors do not change, how much more one can earn if he/she decides to invest $x_1 - x_0$ more dollars.

Model details for experiments

We discuss more details about Bayesian additive Neural Network (AddNN) implementation and provide more experimental results. In our experiments, we use 10 small(sub) neural networks, where each has 2 hidden layers.

Implementation Details: Our Bayesian Neural Network is a sum of 10 small

neural networks and each small network consists of 3 layers. An input feature vector is passed through 10 sub-neural networks followed by addition operation to give a final scalar output. For each sub-neural network, we use 2 to 5 neurons for the first hidden layer and 5 to 20 neurons for the second hidden layer. We train the Bayesian Neural Network with batch size $= 100$ and 0.01 initial learning rate with exponential decay until the validation error converges. In order to pick sparse interpretable variables, we impose group Lasso for the first layer with respect to each input neuron, which associates with sub neural network. The group Lasso penalty hyper-parameter depends on the sparsity and addition structure. In our experiments, it ranges from 0.001 to 1.0.

BANN learns the sparse additive structure: To show our Bayesian additive neural network can learn the sparse addition structure of the function and the interaction, we provide one example of learning Friedman function f_1. We plot the learned matrix of the input layer and the first hidden layer, which can be seen in Fig A.1.

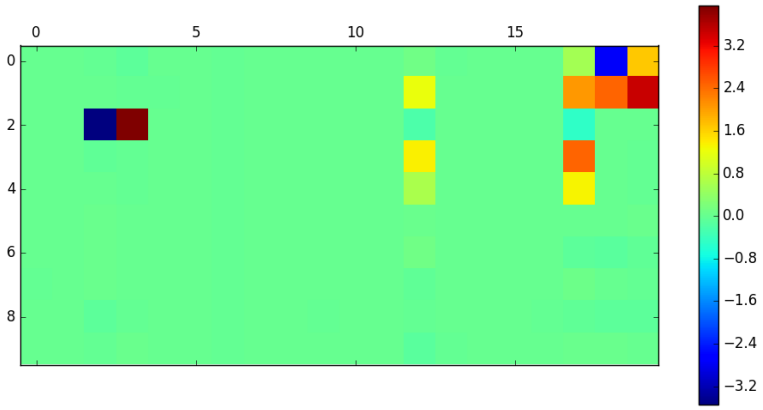


Figure A.1: BANN learns the additive structure of the function f_1. The rows of visualized matrix correspond to the neurons of the input layer and the columns correspond to the 10 sub neural networks, each has 2 hidden units. The sparsity of the matrix demonstrates the sparse interaction between the neurons of the input layer.

REFERENCES

Acid, Silvia, and Luis M De Campos. 1996. An algorithm for finding minimum d-separating sets in belief networks. In *Proceedings of the twelfth uai*.

Aït-Sahalia, Y., J. Cacho-Diaz, and R. J. A. Laeven. 2010. Modeling financial contagion using mutually exciting jump processes. Tech. Rep., National Bureau of Economic Research.

Ancona, Marco, Enea Ceolini, Cengiz ÃƒÂ–ztireli, and Markus Gross. 2018. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International conference on learning representations*.

Ando, Rie Kubota, and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research* 6(Nov):1817–1853.

Angles, TomÃƒÂ¡s, and StÃƒÂ©phane Mallat. 2018. Generative networks as inverse problems with scattering transforms. In *International conference on learning representations*.

Aronszajn, Nachman. 1950. Theory of reproducing kernels. *Transactions of the American mathematical society* 68(3):337–404.

Baktashmotlagh, Mahsa, Mehrtash T Harandi, Brian C Lovell, and Mathieu Salzmann. 2013. Unsupervised domain adaptation by domain invariant projection. In *Proceedings of the iccv*.

Bareinboim, Elias, and Judea Pearl. 2016. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences* 113(27).

Ben-David, Shai, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning* 79(1-2):151–175.

Ben-David, Shai, and Reba Schuller. 2003. Exploiting task relatedness for multiple task learning. In *Learning theory and kernel machines*, 567–580. Springer.

Binkiewicz, N, JT Vogelstein, and K Rohe. 2017. Covariate-assisted spectral clustering. *Biometrika* 104(2):361–377.

Birman, Mikhail Shlemovich, and Mikhail Zakharovich Solomyak. 1967. Piecewise-polynomial approximations of functions of the classes W_p^α. *Matematicheskii Sbornik* 115(3):331–355.

Blei, David M, Alp Kucukelbir, and Jon D McAuliffe. 2017. Variational inference: A review for statisticians. *Journal of the American Statistical Association* 112(518): 859–877.

Boyd, Stephen, and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

Brown, E. N., R. E. Kass, and P. P. Mitra. 2004. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature neuroscience* 7(5):456–461.

Buerger, Katharina, Giovanni Frisoni, Olga Uspenskaya, Michael Ewers, Henrik Zetterberg, Cristina Geroldi, Giuliano Binetti, Peter Johannsen, Paolo Maria Rossini, Lars-Olof Wahlund, et al. 2009. Validation of Alzheimer's disease csf and plasma biological markers: the multicentre reliability study of the pilot european Alzheimer's disease neuroimaging initiative (E-ADNI). *Experimental gerontology* 44(9).

Bui, Thang, Daniel Hernández-Lobato, Jose Hernandez-Lobato, Yingzhen Li, and Richard Turner. 2016. Deep gaussian processes for regression using approximate expectation propagation. In *International conference on machine learning*, 1472–1481.

Carrasco, Marine, and Xiaohong Chen. 2002. Mixing and moment properties of various garch and stochastic volatility models. *Econometric Theory* 18(1):17–39.

Carrillo, Maria C, Kaj Blennow, Holly Soares, Piotr Lewczuk, Niklas Mattsson, Pankaj Oberoi, Robert Umek, Manu Vandijck, Salvatore Salamone, Tobias Bittner,

et al. 2013a. Global standardization measurement of cerebral spinal fluid for Alzheimer's disease: an update from the Alzheimer's association global biomarkers consortium. *Alzheimer's & Dementia* 9(2).

Carrillo, Maria C, Christopher C Rowe, Cassandra Szoeke, Colin L Masters, David Ames, Tim O'Meara, S Lance Macaulay, Andrew Milner, Kathryn A Ellis, Paul Maruff, et al. 2013b. Research and standardization in Alzheimer's trials: reaching international consensus. *Alzheimer's & Dementia* 9(2).

Chatterjee, Soumyadeep, Karsten Steinhaeuser, Arindam Banerjee, Snigdhansu Chatterjee, and Auroop Ganguly. 2012. Sparse group lasso: Consistency and climate applications. In *Proceedings of the 2012 siam international conference on data mining*, 47–58. SIAM.

Chavez-Demoulin, V., and J. A. McGill. 2012. High-frequency financial data modeling using Hawkes processes. *Journal of Banking & Finance* 36(12):3415–3426.

Chen, Jianbo, Le Song, Martin J Wainwright, and Michael I Jordan. 2018. Learning to explain: An information-theoretic perspective on model interpretation. *arXiv preprint arXiv:1802.07814*.

Chen, Xi, Jinghui He, Rick Lawrence, and Jaime G Carbonell. 2012. Adaptive multi-task sparse learning with an application to fmri study. In *Proceedings of the 2012 siam international conference on data mining*, 212–223. SIAM.

Chipman, Hugh A, Edward I George, Robert E McCulloch, et al. 2010. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics* 4(1):266–298.

Cho, Youngmin, and Lawrence K Saul. 2009. Kernel methods for deep learning. In *Advances in neural information processing systems*, 342–350.

Cortes, Corinna, and Mehryar Mohri. 2011. Domain adaptation in regression. In *International conference on algorithmic learning theory*, 308–323. Springer.

Cortes, Corinna, and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20(3):273–297.

Cutajar, Kurt, Edwin V Bonilla, Pietro Michiardi, and Maurizio Filippone. 2017. Random feature expansions for deep gaussian processes. In *International conference on machine learning*, 884–893.

Damianou, Andreas, and Neil Lawrence. 2013. Deep gaussian processes. In *Artificial intelligence and statistics*, 207–215.

Daniely, Amit, Roy Frostig, and Yoram Singer. 2016. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In *Advances in neural information processing systems*, 2253–2261.

Dezeure, Ruben, Peter Bühlmann, Lukas Meier, Nicolai Meinshausen, et al. 2015. High-dimensional inference: Confidence intervals, p-values and r-software hdi. *Statistical Science* 30(4):533–558.

Dezeure, Ruben, Peter Bühlmann, and Cun-Hui Zhang. 2016. High-dimensional simultaneous inference with the bootstrap. *arXiv preprint arXiv:1606.03940*.

Ding, M., CE Schroeder, and X. Wen. 2011. Analyzing coherent brain networks with Granger causality. In *Conf. proc. ieee eng. med. biol. soc.*, 5916–8.

Doukhan, Paul. 1994. *Mixing: properties and examples*. Springer-Verlag.

Dubois, Bruno, Howard H Feldman, Claudia Jacova, Jeffrey L Cummings, Steven T DeKosky, Pascale Barberger-Gateau, André Delacourte, Giovanni Frisoni, Nick C Fox, Douglas Galasko, et al. 2010. Revising the definition of Alzheimer's disease: a new lexicon. *The Lancet Neurology* 9(11).

Efron, Bradley. 2014. Estimation and accuracy after model selection. *Journal of the American Statistical Association* 109(507).

Elwert, Felix. 2013. Graphical causal models. In *Handbook of causal analysis for social research*. Springer.

Fokianos, Konstantinos, Anders Rahbek, and Dag Tjøstheim. 2009. Poisson autoregression. *Journal of the American Statistical Association* 104(488):1430–1439.

Fokianos, Konstantinos, and Dag Tjøstheim. 2011. Log-linear Poisson autoregression. *Journal of Multivariate Analysis* 102(3):563–578.

Fortin, Jean-Michel, and David J Currie. 2013. Big science vs. little science: how scientific impact scales with funding. *PloS one* 8(6).

Gal, Yarin, and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, 1050–1059.

Ganin, Yaroslav, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research* 17(59).

Geer, Sara A. 2000. *Empirical processes in m-estimation*, vol. 6. Cambridge university press.

Gelman, Andrew, and Jennifer Hill. 2006. *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press.

Gong, Boqing, Kristen Grauman, and Fei Sha. 2013. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *Icml*.

Greco, T, A Zangrillo, G Biondi-Zoccai, and G Landoni. 2013. Meta-analysis: pitfalls and hints. *Heart Lung Vessel* 5(4).

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).

Group, Glioma Meta-analysis Trialists GMT. 2002. Chemotherapy in adult high-grade glioma: a systematic review and meta-analysis of individual patient data from 12 randomised trials. *The Lancet* 359(9311):1011–1018.

Gu, Chong. 2013. *Smoothing spline anova models*, vol. 297. Springer Science & Business Media.

Gu, Chong, and Grace Wahba. 1993. Smoothing spline anova with component-wise bayesian "confidence intervals". *Journal of Computational and Graphical Statistics* 2(1):97–117.

Haase, Michael, Rinaldo Bellomo, Prasad Devarajan, Peter Schlattmann, Anja Haase-Fielitz, and NGAL Meta-analysis Investigator Group. 2009. Accuracy of neutrophil gelatinase-associated lipocalin (ngal) in diagnosis and prognosis in acute kidney injury: a systematic review and meta-analysis. *American Journal of Kidney Diseases* 54(6):1012–1024.

Hall, Eric C, Garvesh Raskutti, and Rebecca Willett. 2016. Inference of high-dimensional autoregressive generalized linear models. *arXiv preprint arXiv:1605.02693*.

Hastie, Trevor, and Robert Tibshirani. 1986. Generalized additive models. *Statistical Science* 1(3):297–318.

Heinen, Andréas. 2003. Modeling time series count data: an autoregressive conditional Poisson model. *Available at SSRN 1117187*.

Hernández-Lobato, José Miguel, and Ryan Adams. 2015. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International conference on machine learning*, 1861–1869.

Hosmer, David W, Trina Hosmer, Saskia Le Cessie, Stanley Lemeshow, et al. 1997. A comparison of goodness-of-fit tests for the logistic regression model. *Statistics in medicine* 16(9):965–980.

Huang, Jian, Joel L Horowitz, and Fengrong Wei. 2010. Variable selection in nonparametric additive models. *Annals of statistics* 38(4):2282.

Huang, Jiayuan, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola. 2007a. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*.

Huang, Jiayuan, Alexander J Smola, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, et al. 2007b. Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems* 19:601.

Kim, Seyoung, and Eric P Xing. 2010. Tree-guided group lasso for multi-task regression with structured sparsity. In *Icml*, vol. 2, 1.

Kimeldorf, George, and Grace Wahba. 1971. Some results on tchebycheffian spline functions. *Journal of mathematical analysis and applications* 33(1):82–95.

Klunk, William E, Robert A Koeppe, Julie C Price, Tammie L Benzinger, Michael D Devous, William J Jagust, Keith A Johnson, Chester A Mathis, Davneet Minhas, Michael J Pontecorvo, et al. 2015. The centiloid project: standardizing quantitative amyloid plaque estimation by pet. *Alzheimer's & Dementia* 11(1).

Koltchinskii, V., and M. Yuan. 2010. Sparsity in multiple kernel learning. *Annals of Statistics* 38:3660–3695.

Kontorovich, Leonid. 2007. Measure concentration of strongly mixing processes with applications. Ph.D. thesis, Weizmann Institute of Science.

Kontorovich, Leonid Aryeh, Kavita Ramanan, et al. 2008. Concentration inequalities for dependent random variables via the martingale method. *The Annals of Probability* 36(6):2126–2158.

Lee, Seunghak, Jun Zhu, and Eric P Xing. 2010. Adaptive multi-task lasso: with application to eqtl detection. In *Advances in neural information processing systems*, 1306–1314.

Li, Qi. 2012. Literature survey: domain adaptation algorithms for natural language processing. *Department of Computer Science The Graduate Center, The City University of New York* 8–10.

Lipsey, Mark W.., and David B Wilson. 2001. *Practical meta-analysis*, vol. 49. Sage Publications, Inc.

Liu, Han, Mark Palatucci, and Jian Zhang. 2009. Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery. In *Proceedings of the 26th annual international conference on machine learning*, 649–656. ACM.

Liu, Han, and Jian Zhang. 2009. Estimation consistency of the group lasso and its applications. In *Aistats*, 376–383.

Long, Mingsheng, Yue Cao, Jianmin Wang, and Michael I Jordan. 2015. Learning transferable features with deep adaptation networks. In *Icml*, 97–105.

Mallat, Stéphane. 2016. Understanding deep convolutional networks. *Phil. Trans. R. Soc. A* 374(2065):20150203.

Matteson, David S, Mathew W McLean, Dawn B Woodard, and Shane G Henderson. 2011. Forecasting emergency medical service call arrival rates. *The Annals of Applied Statistics* 1379–1406.

Mattsson, Niklas, Ulf Andreasson, Staffan Persson, Hiroyuki Arai, Sat Dev Batish, Sergio Bernardini, Luisella Bocchio-Chiavetto, Marinus A Blankenstein, Maria C Carrillo, Sonia Chalbot, et al. 2011. The Alzheimer's Association external quality control program for cerebrospinal fluid biomarkers. *Alzheimer's & Dementia* 7(4).

Maurer, Andreas, Massimiliano Pontil, and Bernardino Romera-Paredes. 2013. Sparse coding for multitask and transfer learning. In *Icml (2)*, 343–351.

———. 2016. The benefit of multitask representation learning. *Journal of Machine Learning Research* 17(81):1–32.

Mcdonald, Daniel, Cosma Shalizi, and Mark Schervish. 2011. Estimating beta-mixing coefficients. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 516–524.

Meier, L., S. van de Geer, and P. Buhlmann. 2009. High-dimensional additive modeling. *Annals of Statistics* 37:3779–3821.

Meinshausen, Nicolai, and Bin Yu. 2009. Lasso-type recovery of sparse representations for high-dimensional data. *The Annals of Statistics* 246–270.

Mercer, James. 1909. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character* 209:415–446.

Mnih, Volodymyr, Nicolas Heess, Alex Graves, et al. 2014. Recurrent models of visual attention. In *Advances in neural information processing systems*, 2204–2212.

Mohri, Mehryar, and Afshin Rostamizadeh. 2010. Stability bounds for stationary φ-mixing and β-mixing processes. *Journal of Machine Learning Research* 11(Feb): 789–814.

Mokkadem, Abdelkader. 1988. Mixing properties of arma processes. *Stochastic processes and their applications* 29(2):309–315.

Murdoch, Travis B, and Allan S Detsky. 2013. The inevitable application of big data to health care. *Jama* 309(13):1351–1352.

Nobel, Andrew, and Amir Dembo. 1993. A note on uniform laws of averages for dependent processes. *Statistics & Probability Letters* 17(3):169–172.

Ogata, Y. 1999. Seismicity analysis through point-process modeling: A review. *Pure and Applied Geophysics* 155(2-4):471–507.

Pan, Sinno Jialin, Ivor W Tsang, James T Kwok, and Qiang Yang. 2011. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks* 22(2):199–210.

Patel, Vishal M, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. 2015. Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine* 32(3):53–69.

Pearl, Judea. 2014. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier.

Raskutti, Garvesh, Martin J Wainwright, and Bin Yu. 2012. Minimax-optimal rates for sparse additive models over kernel classes via convex programming. *Journal of Machine Learning Research* 13(Feb):389–427.

Rasmussen, Carl Edward. 2003. Gaussian processes in machine learning. In *Summer school on machine learning*, 63–71. Springer.

Ravikumar, P., H. Liu, J. Lafferty, and L. Wasserman. 2010. SpAM: sparse additive models. *Journal of the Royal Statistical Society, Series B*.

Rudelson, Mark. 2008. Invertibility of random matrices: norm of the inverse. *Annals of Mathematics* 575–600.

Rydberg, Tina Hviid, and Neil Shephard. 1999. A modelling framework for the prices and times of trades made on the new york stock exchange. Tech. Rep., Nuffield College. Working Paper W99-14.

Sabour, Sara, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *Advances in neural information processing systems*, 3859–3869.

Saitoh, Saburou. 1988. *Theory of reproducing kernels and its applications*, vol. 189. Longman.

Salimbeni, Hugh, and Marc Deisenroth. 2017. Doubly stochastic variational inference for deep gaussian processes. In *Advances in neural information processing systems*, 4591–4602.

Shaw, Leslie M, Hugo Vanderstichele, Malgorzata Knapik-Czajka, Christopher M Clark, Paul S Aisen, Ronald C Petersen, Kaj Blennow, Holly Soares, Adam Simon, Piotr Lewczuk, et al. 2009. Cerebrospinal fluid biomarker signature in Alzheimer's disease neuroimaging initiative subjects. *Annals of neurology* 65(4).

Shrikumar, Avanti, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *International conference on machine learning*, 3145–3153.

Simon, Noah, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2013. A sparse-group lasso. *Journal of Computational and Graphical Statistics* 22(2):231–245.

Smola, Alex J, and Bernhard Schölkopf. 1998. *Learning with kernels*. GMD-Forschungszentrum Informationstechnik.

Stegenga, Jacob. 2011. Is meta-analysis the platinum standard of evidence? *Studies in history and philosophy of biological and biomedical sciences* 42(4).

Stone, C. J. 1985. Additive regression and other nonparametric models. *Annals of Statistics* 13(2):689–705.

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, 3319–3328.

Tao, Terence. 2012. *Topics in random matrix theory*, vol. 132. American Mathematical Soc.

Tarca, Adi L, Vincent J Carey, Xue-wen Chen, Roberto Romero, and Sorin Drăghici. 2007. Machine learning and its applications to biology. *PLoS Comput Biol* 3(6):e116.

Tian, Jin, Azaria Paz, and Judea Pearl. 1998. *Finding minimal d-separators*. Citeseer.

Tsang, Michael, Dehua Cheng, and Yan Liu. 2018. Detecting statistical interactions from neural network weights. In *International conference on learning representations*.

Vanderstichele, Hugo, Mirko Bibl, Sebastiaan Engelborghs, Nathalie Le Bastard, Piotr Lewczuk, Jose Luis Molinuevo, Lucilla Parnetti, Armand Perret-Liaudet, Leslie M Shaw, Charlotte Teunissen, et al. 2012. Standardization of preanalytical aspects of cerebrospinal fluid biomarker testing for Alzheimer's disease diagnosis: a consensus paper from the Alzheimer's biomarkers standardization initiative. *Alzheimer's & Dementia* 8(1).

Verwey, NA, WM Van Der Flier, K Blennow, C Clark, S Sokolow, PP De Deyn, Douglas Galasko, Heather Hampel, T Hartmann, Elisabeth Kapaki, et al. 2009. A worldwide multicentre comparison of assays for cerebrospinal fluid biomarkers in Alzheimer's disease. *Annals of clinical biochemistry* 46(3).

Vidyasagar, Mathukumalli. 2002. *A theory of learning and generalization*. Springer-Verlag New York, Inc.

Wager, Stefan, and Susan Athey. 2017. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*.

Wager, Stefan, Trevor Hastie, and Bradley Efron. 2014. Confidence intervals for random forests: the jackknife and the infinitesimal jackknife. *Journal of Machine Learning Research* 15(1).

Wahba, Grace. 1990. *Spline models for observational data*. SIAM.

Wang, HaiYing, Rong Zhu, and Ping Ma. 2017. Optimal subsampling for large sample logistic regression. *Journal of the American Statistical Association*.

Wang, Li-San, Yuk Yee Leung, Shu-Kai Chang, Susan Leight, Malgorzata Knapik-Czajka, Young Baek, Leslie M Shaw, Virginia M-Y Lee, John Q Trojanowski, and Christopher M Clark. 2012. Comparison of xMAP and ELISA assays for detecting cerebrospinal fluid biomarkers of Alzheimer's disease. *Journal of Alzheimer's Disease* 31(2).

Weinert, Howard L. 1982. *Reproducing kernel hilbert spaces: applications in statistical signal processing*, vol. 25. Hutchinson Ross Pub. Co.

Wilson, Andrew Gordon, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. 2016. Deep kernel learning. In *Artificial intelligence and statistics*, 370–378.

Xu, Kelvin, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, 2048–2057.

Yu, Bin. 1994. Rates of convergence for empirical processes of stationary mixing sequences. *The Annals of Probability* 94–116.

Zeiler, Matthew D, and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833. Springer.

Zhang, Yilin, Marie Poux-Berthe, Chris Wells, Karolina Koc-Michalska, and Karl Rohe. 2018. Discovering political topics in facebook discussion threads with graph contextualization. *The Annals of Applied Statistics* 12(2):1096–1123.

Zhao, Tuo, and Han Liu. 2012. Sparse additive machine. In *Aistats*, 1435–1443.

Zhou, Hao Henry, and Garvesh Raskutti. 2019. Non-parametric sparse additive auto-regressive network models. *IEEE Transactions on Information Theory* 65(3): 1473–1492.

Zhou, Hao Henry, Sathya N Ravi, Vamsi K Ithapu, Sterling C Johnson, Grace Wahba, and Vikas Singh. 2016. Hypothesis testing in unsupervised domain adaptation with applications in Alzheimer's disease. In *Advances in neural information processing systems 29*, 2496–2504.

Zhou, Hao Henry, Vikas Singh, Sterling C Johnson, Grace Wahba, and the Alzheimer's Disease Neuroimaging Initiative. 2018a. Statistical tests and identifiability conditions for pooling and analyzing multisite datasets. *Proceedings of the National Academy of Sciences* 115(7):1481–1486.

Zhou, Hao Henry, Yunyang Xiong, and Vikas Singh. 2018b. Building bayesian neural networks with blocks: On structure, interpretability and uncertainty. *arXiv preprint arXiv:1806.03563*.

Zhou, Hao Henry, Yilin Zhang, Vamsi K Ithapu, Sterling C Johnson, Grace Wahba, and Vikas Singh. 2017. When can multi-site datasets be pooled for regression? Hypothesis tests, ℓ_2-consistency and neuroscience applications. In *Proceedings of the 34th international conference on machine learning*, vol. 70, 4170–4179.

Zhou, K., H. Zha, and L. Song. 2013. Learning social infectivity in sparse low-rank networks using multi-dimensional Hawkes processes. In *Proceedings of the 16th international conference on artificial intelligence and statistics (aistats)*.