ADVANCED ANALYSIS METHODS FOR LARGE-SCALE STRUCTURED DATA

Fan Zhou

A dissertation submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Biostatistics in the Gillings School of Global Public Health.

Chapel Hill
2019

Approved by:

Hongtu Zhu

Haibo Zhou

Lisa Lavange

Fei Zou

Yen-Yu Ian Shih

ABSTRACT

Fan Zhou: Advanced Analysis Methods For Large-Scale Structured Data
(Under the direction of Hongtu Zhu and Haibo Zhou)

In the era of 'big data', advanced storage and computing technologies allow people to build and process large-scale datasets, which promote the development of many fields such as speech recognition, natural language processing and computer vision. Traditional approaches can not handle the heterogeneity and complexity of some novel data structures. The target of this dissertation is to develop new statistical models to solve all kinds of real-world problems based on structured data from different areas.

Three different data structures are discussed in this dissertation. In the first part of the dissertation, we introduce a novel data sampling scheme: muti-group association data, which is widely adopted by recent medical studies with multi-class disease outcomes. We develop a general regression framework for the secondary phenotype analysis using multi-group data to correct the estimation bias caused by the uneven sampling rates of different sub-groups.

The second data type being included is the graph-based data, i.e. the network data. In this dissertation, we discuss the graph-based semi-supervised learning problem with nonignorable missingness, which is ignored by most previous studies. We do both simulation and real analysis using citation networks to show the necessity of doing bias correction when there exists nonignorable nonresponses.

Prediction of customer requests with both origin and destination locations in the future is a fundamental question to the ride-sharing systems. In the last chapter of the dissertation, we propose a deep-learning based model to jointly capture the spatial-temporal features of this kind of Origin-Destination (OD) networks and make predictions for the flow values in

the incoming time window given the historical information. Some experiments using the demand data from DiDi demonstrates the advantage of our model in predicting OD flow data in practice.

To my mentor, parents and friends, I couldn't have done this without you. Thank you for all of your support along the way.

ACKNOWLEDGEMENTS

I would like to thank my advisors, Dr. Hongtu Zhu and Dr. Haibo Zhou, for your mentoring through my whole PhD career. Your guidance has helped nurture my independent thinking, academic sense, passion in exploring unknown fields, as well as self-learning and collaboration ability.

I would also like to thank Dr. Lisa Lavange, Dr. Fei Zou and Dr. Yen-Yu Ian Shih for serving as my committee members. Your valuable comments and suggestions really help.

To family members, friends and my dear fiancee, I can not finish my PhD thesis without your endless love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

By three different research topics, we explore how to combine useful tools, including both traditional approaches and deep learning architectures, to develop new methodologies in analyzing certain kinds of structured data. The first two topics try to correct the estimation bias that results from unusual sampling designs. The latter two deal with some real-world problems people are interested in generated by network data.

Multi-group design, such as the Alzheimer's Disease Neuroimaging Initiative (ADNI), has been undertaken by recruiting subjects based on their multi-class primary disease status, while some extensive secondary outcomes are also collected. Analysis by standard approaches is usually distorted because of the unequal sampling rates of different classes. In the first part of the dissertation, we develop a general regression framework for the analysis of secondary phenotypes collected in multi-group association studies. Our regression framework is built on a conditional model for the secondary outcome given the multi-group status and covariates and its relationship with the population regression of interest of the secondary outcome given the covariates. Then, we develop generalized estimation equations to estimate the parameters of interest. We use simulations and a large-scale imaging genetic data analysis of the ADNI data to evaluate the effect of the multi-group sampling scheme on standard genomewide association analyses based on linear regression methods, while comparing it with our statistical methods that appropriately adjust for the multi-group sampling scheme.

In the past few decades, network data has been increasingly collected and studied in diverse areas, including neuroimaging, social networks and knowledge graphs. In the second part of the dissertation, we investigate the graph-based semi-supervised learning problem with nonignorable nonresponses. We propose a Graph-based joint model with Nonignorable Missingness (GNM) and develop an imputation and inverse probability weighting estimation

approach. We further use graph neural networks (GNN) to model nonlinear link functions and then use a gradient descent (GD) algorithm to estimate all the parameters of GNM. We propose a novel identifiability for the GNM model with neural network structures, and validate its predictive performance in both simulations and real data analysis through comparing with models ignoring or misspecifying the missingness mechanism. Our method can achieve up to 7.5% improvement than the baseline model for the document classification task on the Cora dataset.

Predictions of Origin-Destination (OD) flow data is an important instrument in transportation studies. However, most existing methods ignore the network structure of OD flow data. In the last part of the dissertation, we propose a spatial-temporal origin-destination (STOD) model, with a novel CNN filter to learn the spatial features from the perspective of graphs and an attention mechanism to capture the long-term periodicity. Experiments on a real customer request dataset with available OD information from a ride-sharing platform demonstrates the advantage of STOD in achieving a more accurate and stable prediction performance compared to some state-of-the-art methods.

CHAPTER 2: LITERATURE REVIEW

In this chapter, we review some existing representative works related to the topics covered in this dissertation. In section 1.1, we introduce three unusual data types: muti-group association data, graph-based data and origin-destination flow data. We briefly discuss the research problems people are interested in and the accompanying statistical challenges when analyzing these three kinds of structured data. In section 1.2, we review a large set of literature on the development of statistical methods to eliminate the selection bias related to ascertainment in case-control studies for secondary trait analysis. In section 1.3, we review the main approaches to obtain unbiased parameter estimations in the presence of nonignorable missingness. In section 1.4, we go through the developing history of prediction models applied to dynamic spatial-temporal data.

2.1 Structured Data

2.1.1 Multi-Group Association Data

Case-control (Cornfield, 1951) is a special design of observational study, which recruits two groups of people with potentially different outcomes to certain diseases to explore their association with some exposure variables of interest. The case-control study follows a retrospective design since the primary outcome of each individual is known before it being enrolled and all the covariate information can be retrieved.

Case-control studies have several advantages over traditional sampling mechanisms. Randomly selecting subjects from the whole population requires a larger sample size to significantly discriminate the cases from controls especially in the rare disease case, which results in inefficient data utilization. Case-control design addresses this issue by oversampling the cases and hiring a matched number of control subjects. White (1982) extends case-control to a two-stage situation, and demonstrates its advantage over one-stage design. The two stages

follow different sampling schemes, where the first stage is equivalent to a standard case-control sample and subjects in the second stage are subdivided into four groups: two case groups (diseased and exposed/unexposed) and two control groups (normal and exposed/unexposed). The two-stage design is more efficient and flexible because the sample sizes of the four subgroups can vary with the disease and exposure rates. Breslow and Cain (1988) propose an irregular logistic regression for the two-stage case-control design, the efficiency of which is maximized when the exposure rate is rare. Flanders and Greenland (1991) introduces a pseudo-likelihood approach to analyze the data acquired from two-stage case-control studies.

Although case-control design has been widely used in biological studies, they are insufficient for many complex diseases, such as Alzheimer's disease and breast cancer. These diseases may have multiple subtypes with distinct morphologies and clinical implications. To recruit enough people for each disease subtype, multi-group design can be employed to sample subjects within different groups in different proportions from the whole population. One typical example following the multi-group design is the Alzheimer's Disease Neuroimaging Initiative (ADNI), which has three main groups: Alzheimer's disease (AD), mild cognitive impairment (MCI), and elderly controls (NC). The major goal of ADNI data set is to promote the development of longitudinal, multi-site, imaging-genetic methods in analyzing Alzheimer's disease. Patients from the three groups are non-randomly sampled with different probabilities where a total of 800 subjects including 200 normal controls, 400 individuals with MCI, and 200 subjects with mild AD are recruited by ADNI1. More than 50% subjects in the sample are with MCI since researchers want to explore more about the transition mechanism from MCI to AD while no more than 15% of people older than 55 are in MCI status in the whole population. ADNI has gone though four phases from ADNI1, GO, 2 to ADNI3 from 2004 until 2016 and the whole sample size is extended to over 1700. A new cohort Significant Memory Concern (SMC) is added since ADNI2.

Another field multi-group design being widely used is the cancer study. Wang et al. (2017) discusses a tissue microarray (TMA) imaging dataset for thyroid cancer. Patients who had

surgery for thyroid cancer at Mackay Memorial Hospital between January 2001 and May 2012 are recruited to build the sample. The TMA data is usually generated by the tissue sections cut from both normal and tumor samples. The proportion of subjects in the sample with more severe cancer stages are much higher than those in the whole population, which makes the TMA dataset a non-random sample.

Similar to the case-control design, estimations by standard models using the muti-group association data can be extremely misleading. In this dissertation, we build a general framework to properly correct the sampling bias when analyzing secondary phenotypes in multi-group association studies.

2.1.2 Graph-Based Data

Graphs can be used to represent either symmetric or asymmetric relations between a group of discrete objects. With technology development and population growth, large-scale graph-based datasets are generated to solve all kinds of real-world problems.

Graph-based semi-supervised learning problem has been increasingly studied, the goal of which is to predict the node responses of all the unlabelled vertexes (such as documents) in a graph (such as a citation network) based on only a small subset of observed ones. The labelling information is usually smoothed over the graph via some form of explicit graph-based regularization.

A popular method is to use the graph Laplacian regularization to learn node representations, such as label propagation (Zhu et al., 2003), manifold regularization (Belkin et al., 2006) and deep semi-supervised embedding (Weston et al., 2012).

Recently, attention has been shifted to the learning of network embeddings, which is first discussed in skip-gram model (Mikolov et al., 2013). Perozzi et al. (2014) presents Deep-Walk to learn the latent representations of vertices in a network using local information obtained from truncated random walks. LINE (Tang et al., 2015) and node2vec (Grover and Leskovec, 2016) improve Deep-Walk by allowing more flexibility when exploring neighborhoods through random walks. However, all these methods are based on a multi-step framework,

where the generation of random walks and the main semi-supervised classifier are built and optimized individually. Yang et al. (2016) proposes a novel graph-based semi-supervised learning framework. Different from the above two-step procedures, the network embedding and the final classification model are jointly trained by an end-to-end architecture. The graph embedding and hidden representation learned from the classifier are concatenated to feed into the final prediction layer.

In the past few years, more efforts have been devoted to developing deep learning models to capture the spatial information of network data (Bruna et al., 2013; Henaff et al., 2015; Duvenaud et al., 2015; Li et al., 2015). They either pay attention to problem-specific specialized architectures or utilize graph convolutions known as spectral graph theory. Defferrard et al. (2016) designs a localized convolution network for general graph structures. The lower layers of the network is convolutional in the sense that the same local filter is applied to each graph vertex and its neighboring nodes. Then a global pooling procedure combines the features captured by a multi-layer propagation from all the vertexes.

We consider a weighted graph structure consisting of an undirected (or directed) graph $G = (V, E)$ as well as an adjacency matrix $A = (a_{ij})$, where a_{ij}s' are nonegative edge weights, $V = \{v_1, \ldots, v_N\}$ is a set of $|V| = N$ vertices, and E is a set of edges. Moreover, $(v_i, v_j) \in E \subset V \times V$ is an edge equipped with an nonegative weight a_{ij}. The adjacency matrix $A = (a_{ij}) \in R^{N \times N}$ encodes the node connections. $x \in R^{N \times c}$ is the node-level signals where c is the length of feature vectors.

aAn widely-used operator in spectral graph analysis is the graph Laplacian (Chung and Graham, 1997). Formally, the graph Laplacian given the adjacency matrix A is defined as $L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ where $D \in R^{N \times N}$ is a diagonal degree matrix with $D_{ii} = \sum_j a_{ij}$. $L = U^T \Lambda U$ is the eigenvalue decomposition of L with U being the matrix of eigenvectors and $\Lambda = \text{diag}([\lambda_0, \lambda_1, \ldots, \lambda_{N-1}])$ being the diagnal matrix containing eigenvalues. We consider spectral convolutions on graphs defined as the multiplication of the input matrix $x \in R^{N \times c}$ with a filter g_θ in the Fourier domain (Defferrard et al., 2016). The filter g_θ serves as the

function of the eigenvalues of L, i.e. $g_\theta(\Lambda)$. Hammond et al. (2011) suggests that $g_\theta(\Lambda)$ can be well-approximated by a truncated expansion in terms of Chebyshev polynomials $T_k(x)$ up to K-th order:

$$g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k T_k(\Lambda) \tag{2.1}$$

The Chebyshev polynomials are recursively defined as $T_k(z) = 2zT_{k-1}(z) - T_{k-2}(z)$ with $T_0 = 1$ and $T_1 = z$.

The spectral graph convolutions at the l-th layer incorporated with input $m_t^l \in R^{N \times c_l}$ can be modified as:

$$g_\theta * m_t^l \approx \sum_{k=0}^{K-1} T_k(\tilde{L}) m_t^l W_l \tag{2.2}$$

where $\tilde{L} = \frac{2}{\lambda_{\max}} L - I_N$ with $\lambda_{\max}$ being the maximum eigenvalue of the Laplacian matrix. $W_l \in R^{c_l \times d}$ is the GCN projection matrix to learn. Assuming $\tilde{x}_k = T_k(\tilde{L})x$, by the recurrence relations we have $\tilde{x}_k = 2\tilde{L}\tilde{x}_{k-1} - \tilde{x}_{k-2}$ with $\tilde{x}_0 = x$ and $\tilde{x}_1 = \tilde{L}x$.

Kipf and Welling (2016) simplify the graph convolution networks proposed by Defferrard et al. (2016) to highly increase the training efficiency and obtain a higher prediction accuracy. The layer-wise transformation is defined as:

$$f(H^{(l)}, A) = \sigma(AH^{(l)}W^{(l)}) \tag{2.3}$$

where $W^{(l)}$ is a weight matrix for the l-th layer and $\sigma(\cdot)$ is a non-linear activation function such as the ReLU. $H(0) = X$ serves as the input and $H(L) = Z$ is the final output when there are in total L layers. Despite the simple structure the proposed operation, the model is powerful in capturing the graph-based spatial information.

There are two main limitations of the operation above. One is the multiplication of A at each layer, which models the spatial information of neighboring nodes but dismisses the target node itself unless self-loops exist in the graph. A simple solution to solve this problem is to add an identity matrix to A.

Another limitation is that A is not normalized and multiplication with A will keep changing the scale of the output representations at each layer. Therefore, Kipf and Welling (2016) normalize A to make the row sums to be one, i.e. $D^{-1}A$, where D is the diagonal matrix summing up each row of A. Multiplying with $D^{-1}A$ is equivalent to take a weighted sum over the neighboring grids and the center grid itself. A more advanced way is to use a symmettic normalization $D^{-1/2}AD^{-1/2}$ (as this no longer amounts to mere averaging of neighboring nodes). With the normalization of the the mutliplication, (2.3) is modified to

$$f(H^{(l)}, A) = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \tag{2.4}$$

where $\hat{A} = A + I$, where I is the identity matrix and $\hat{D}$ is the diagonal node degree matrix of $\hat{A}$. The propagation rule could be seen as the first-order approximation of localized spectral filters on graphs (Defferrard et al., 2016).

2.1.3 OD flow data

Spatial-temporal prediction of large-scale OD flow networks plays an important role in traffic flow control, urban routes planning, infrastructure construction, and policy design of ride-sharing platforms, among others. On ride-sharing platforms, customers keep sending requests with origins and destinations at each moment. Knowing the exact original location and destination of each future trip allows platforms to prepare sufficient supplies in advance to optimize resource utilization and improve users' experience. Given the destinations of prospective demands, platforms can predict the number of drivers transferring from busy to idle status. Prediction of dynamic demand flow data helps ride-sharing platforms to design better order dispatch and fleet management policies for achieving the demand-supply equilibrium as well as decreased passenger waiting times and increased driver serving rates.

There is a great interest in building spatial-temporal models to predict the total number of customer requests at each origin-destination pair generated in the $(t + j)$-th time interval, given the historical demand data until the current time window t. We consider the set of

dynamic OD flow maps as a sequence of graph snapshots $G = \{G_1, \ldots, G_T\}$. With the OD flow network at each time $t \in \{1, \ldots, T\}$, we can define a weighted graph $G_t = (V, O_t)$ with a fixed vertex set $V = \{v_1, \ldots, v_N\}$ representing $|V| = N$ urban regions. The dynamic adjacency matrix $O_t = (o_t^{ij}) \in R^{N \times N}$ describes flow amounts within all N^2 OD flows, where o_t^{ij} represents the flow amount from node v_i to node v_j at timestamp t.

Many efforts have been devoted to developing traffic flow prediction models in the past few decades. Before the rise of deep learning, traditional statistical and machine learning approaches dominate this field. These methods are usually built on linear transformations, so they often ignore non-linear correlations among the OD flows. Some other methods further use additional external features obtained from feature engineering, but they fail to automatically extract the spatial representation of OD data. Moreover, they roughly combine the spatial and temporal features when fitting the prediction model instead of dynamically model their interactions.

The development of deep learning technologies brings a significant improvement of OD flow prediction by extracting non-linear latent structures that cannot be easily discovered by feature engineering. For instance, convolutional operations are often used to capture more complicated spatial patterns in the OD flow data, most of which treat each O_t as an image. In this case, some nearby OD flows in O_t covered by a single CNN kernel may not be semantically correlated. On the other hand, two neighboring OD flows with shared vertexes in the graph can be far from each other in terms of images. As we mentioned in the previous section, graph-based neural networks (GNN) (Kipf and Welling, 2016; Defferrard et al., 2016) are proved to be powerful tools for modelling network structures. However, none of them are directly applicable here since both the input and output of GNNs are node-level features. For OD flow prediction problems, the spatial information in edge space is more important because of the equivalence between OD flows and graph edges by our definition.

2.2 Secondary Phenotype Analysis

In this section, we review the existing methods for secondary phenotype analysis. We will focus on the case-control design since almost all the existing methods are designed for the two-group situation, where both the binary disease status and some secondary phenotypes are collected.

In case-control studies, subjects of disease and control groups are selected with different probabilities from the whole population. Therefore, fitting a standard regression model is statistically biased when analyzing the secondary phenotypes. There are several ways to correct the estimation bias caused by the uneven sampling rates of the two groups. Before moving to the details of these approaches, we introduce some important notations first. Let D be the primary binary outcome (case-control status) and Y be the secondary outcome (which could be either continuous or categorical). X denotes the set of covariates to analyze. The simplest method is to fit a standard regression model using a subset of observations. All these naive approaches can fall into four broad categories depending on the groups of subjects being included:

1. Regress Y over X using control subjects only.

2. Regress Y over X using case subjects only.

3. Regress Y over X using the entire sample.

4. Include the case-control status D as an additional covariate in the regression models.

However, none of these approaches are statistically correct. (1) and (2) require a strong assumption that there exists no significant group difference regarding the covariate effects onto the target secondary phenotypes. Moreover, dropping a certain number of observations can substantially decrease the estimation efficiency and statistical power. (3) is another naive approach which treats the case-control sample as a random sample from the whole population. Jiang et al. (2006); Lin and Zeng (2009); Monsees et al. (2009) point out that (3) is valid if

and only if $Y \perp D|X$. (4) may yield flawed conclusions, since the associations between the secondary outcome and an exposure of interest in the case and control groups can be quite different from that in the underlying target population (Tchetgen Tchetgen, 2014).

Faced with the increasing demand in analyzing secondary traits on case-control sample, a number of well-designed modified statistical approaches are proposed. All these methods can be roughly divided into three main classes: (1) Inverse Probability Weighting (IPW) methods. (2) Likelihood-based methods. (3) Semiparametric efficient estimating methods.

2.2.1 Inverse Probability Weighting

Various weighted likelihood approaches, such as the inverse probability weighting (IPW), have been widely used (Richardson et al., 2007; Monsees et al., 2009; Schifano et al., 2013; Sofer et al., 2017) to correct sampling bias. The IPW-based approaches replace the normal log-likelihood function by a weighted sum using weights w_i given by the reciprocal of the selection probability for each subject in the case-control sample. We let the target of inference be $f_\beta(Y|X)$ with β including all the parameters related to the the conditional mean model. If the total sample size is N, the weighted log-likelihood function is defined as:

$$l(\beta) = \sum_{i=1}^{N} \frac{1}{w_i} \log f_\beta(Y_i|X_i) \tag{2.5}$$

which is proved to provide unbiased estimation of β and appropriate type-one error rates.

Schifano et al. (2013) extends the IPW approach to multiple-response situation, improving the statistical power by borrowing strength across outcomes with a one degree of freedom test and jointly estimating the outcome-specific exposure effects when the secondary phenotypes are positively correlated. Suppose $y_i = (y_{i1}, \ldots, y_{iM})$ denotes the M-dimension correlated continuous phenotypes and σ_i^2 being the phenotype-specific variance, the weighted estimating equations is defined as:

$$\sum_{i=1}^{N} w_i \mathbf{X}_i^T R^{-1} (\frac{y_i}{\sigma_i} - \mathbf{X}_i \gamma) = 0 \tag{2.6}$$

and

$$\sum_{i=1}^{N} w_i \{ \frac{y_{ij}}{\sigma_j} (\frac{y_{ij}}{\sigma_j} - \mathbf{x}_i^T \beta) - 1 \} = 0, \quad j = 1, \ldots, M \tag{2.7}$$

where R is the working correlation matrix and

$$\mathbf{X}_i = \begin{bmatrix} \mathbf{x}_i^T & 0^T & \ldots & 0^T \\ 0^T & \mathbf{x}_i^T & \ldots & 0^T \\ \vdots & & \ddots & \vdots \\ 0^T & 0^T & \ldots & \mathbf{x}_i^T \end{bmatrix}$$

The weight w_i equals to the global prevalence divided by the sample-level group proportions. Schifano et al. (2013) proves that the proposed estimating equation is unbiased.

Sofer et al. (2017) points out that IPW is inefficient because of ignoring the data generating mechanism. To address this issue, they propose a novel class of estimators which combine traditional IPW with specification of the disease outcome probability model via a mean zero control function. The control-function assisted IPW estimating equations is defined as follows:

$$U(\beta) = \sum_{i=1}^{N} \frac{1}{\pi(D_i)} (h_1(X_i)[Y_i - g^{-1}\{\mu(X_i;\beta)\}] - h_2(X_i, D_i)) = 0 \tag{2.8}$$

where $\pi(D_i) = Pr(S_i = 1 | X_i, D)$ and $[Y|X] = g^{-1}\{\mu(X;\beta)\}$ are the population-level conditional mean model. S_i is a binary variable indicating whether a subject is selected into the sample. $h_1(X, D), h_2(X, D)$ are the control functions which depend on the disease model and satisfies $\{h_2(X, D)/\pi(D) | X, S = 1\} = 0$. In this case, the inverse probability weight becomes $h_2(X, D)/\pi(D)$ with mean zero sum.

In practice, IPW-based methods are usually inefficient since some information related to D is not fully utilized. Likelihood-based and semiparametric estimation approaches could solve this problem to some extent, the details of which are discussed in the following subsections.

2.2.2 Likelihood-based Methods

Lee et al. (1997) develops a maximum likelihood estimating equation to jointly model the

conditional distribution of D and Y given X when the sampling rates for the two groups are known. Jiang et al. (2006) carries out an extensive investigation of efficiency and proves that the semi-parametric maximum likelihood methods are theoretically more efficient than the weighted likelihood methods.

Lin and Zeng (2009) introduces a retrospective likelihood function by explicitly conditioning on the sampling scheme. If Y is a continuous outcome, a linear regression model could be used when assuming Y given X follows a normal distribution with mean $\beta_0 + \beta_1 X$ and variance σ^2. When Y is the binary outcome, we model $Y|X$ by a logistic regression:

$$P(Y = 1|X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}} \tag{2.9}$$

Moreover, another logistic regression is used to describe the relationship between D and (Y, X) as:

$$P(D = 1|X, Y) = \frac{e^{\gamma_0+\gamma_1 X+\gamma_2 Y}}{1 + e^{\gamma_0+\gamma_1 X+\gamma_2 Y}} \tag{2.10}$$

Because the sampling is conditional on the case-control status, the likelihood function takes the retrospective form:

$$\prod_{i=1}^{N} \left\{ \frac{P(D_i = 1|X_i, Y_i)P(Y_i|X_i)P(X_i)}{P(D_i = 1)} \right\}^{D_i} \left\{ \frac{P(D_i = 0|X_i, Y_i)P(Y_i|X_i)P(X_i)}{P(D_i = 0)} \right\}^{1-D_i} \tag{2.11}$$

where $P(D_i = 1) = \sum_y \sum_x P(D_i = 1|x, y)P(y|x)P(x)$, $P(D_i = 1) = 1 - P(D_i = 0)$. Lin and Zeng (2009) proposes a profile-likelihood approach to eliminate the nuisance parameters from the potential high-dimensional probability distribution of continuous environmental covariates. Specifically, they treat the distribution of x as discrete point masses $p_i = p(x_i)$ based on the N finite observations in the case-control sample. $\sum_{i=1}^{N} p_i = 1$ is the added additional constraint when maximizing the objective likelihood function. According to simulation results, their method provides an unbiased estimation, accurately controlling the type-one error and maximizing the statistical power.

He et al. (2012) uses a gaussian copula approach, allowing more flexible distributions of the secondary outcome Y compared to Lin and Zeng (2009), which works for the multiple-outcome case.

2.2.3 Semiparametric and Estimating Equation Methods

Wei et al. (2013) proposes a robust estimation method for secondary analysis of case-control data by assuming that the secondary trait Y given X follows a homoscedastic regression model, which is defined as

$$Y = \alpha + \mu(X, \beta) + \epsilon \tag{2.12}$$

where α is the intercept and μ is a known function. ϵ is the zero-mean error term which is independent of X.

The method by Wei et al. (2013) allows the model for Y given X to be incorrect, and makes the estimation approach robust. One main assumption of this method is that the disease rate is given or could be well estimated. They pursue a sequential approach to estimate the parameters related to the target regression model $Y|X$. The details of the algorithm are described in three steps as follows:

1. Estimate the logistic regression of D given (X, Y) and obtain the related parameters κ, θ_1. The logistic model is defined as:

$$P(D = 1|X, Y) = \frac{e^{\theta_0 + m(Y,X;\theta_1)}}{1 + e^{\theta_0 + m(Y,X;\theta_1)}} \tag{2.13}$$

   On the other hand, $\kappa = \theta_0 + \log(n_1/n_0) - \log(\pi_1/\pi_0)$ where n_1, n_0 are the number of subjects in case and control group, and π_1, π_0 are global prevalences for the two groups in the whole population. Prentice and Pyke (1979); Chatterjee and Carroll (2005) demonstrate that θ_1 and κ can be consistently estimated by the standard logistic regression using the case-control sample. Moreover, it is assumed that a consistent estimation of θ_0 could also be obtained by solving an estimating equation.

2. Define a proper score function for β when (Y, X) are randomly sampled from the whole

population. The simplest way to acquire the score function is to take the derivative of the ordinary least squares $\{Y - \alpha - \mu(X, \beta)\}^2$, making the score funtion to be

$$L\{R(\beta), X, \alpha, \beta\} = \mu_\beta(X, \beta)\{R(\beta) - \alpha\} \tag{2.14}$$

where $R(\beta) = Y - \mu(X, \beta)$. The score (2.14) is then adjusted to have zero-mean under case-control design.

3. Denote $\Omega = (\kappa, \theta_0, \theta_1)$ and replace α in the score function by $\alpha(\beta, \Omega)$. Solve the adjusted score equation and get the estimation of β and hence α.

Song et al. (2016) introduces a set of counter-factual estimation functions under an alternative disease status, and combines the observed and counter-factual estimation functions into a set of weighted estimation equations (WEE). Simulations results demonstrates that WEE is more robust against biased sampling and less sensitive to model misspecification.

Assuming $S(X, Y, \beta)$ is an estimating function with $E_Y(X, Y, \beta^*)|X)$ at true value β^*, the unbiased counterfactual estimating equation by conditional expectation is defined as:

$$S_n(\beta) = \sum_{i=1}^{N} [S(x_i, y_i, \beta)p(d_i|x_i) + E_{\tilde{y}_i}[S(x_i, \tilde{y}_i, \beta)|x_i]p(1 - d_i|x_i)] = 0 \tag{2.15}$$

where y_i is the observation in the sample and $\tilde{y}_i$ is the counter-factual secondary outcome under the alternative disease status. Estimating equation (2.15) remains unbiased when $S(x_i, \tilde{y}_i, \beta)$ is non-linear. Another estimation approach is to fit the model $Y|X$ for cases and controls separately, and then generate pseudo counter-factual observations using the resulting stratified models.

Ma and Carroll (2016) constructs a class of semiparametric estimation procedures which does not rely on a fully parametric distributions of the error term, specified disease rates or an approximation in the whole population. Only the regression mean model is specified while the error term can be heteroscedastic and depend on the covariates. The Regression model

of Y given X in the whole population is defined as:

$$Y = m(X, \beta) + \epsilon \tag{2.16}$$

where $m(\cdot)$ is a known function and ϵ is the zero-mean error term. To relax the assumptions of error distribution and disease rates, the concept of a superpopulation (Ma et al., 2010) is adopted. Under the superpopulation framework, the regression model can be rewritten as:

$$f^{\text{true}}_{Y|X}(X, y) = \eta_2\{y - m(X, \beta), X\} \tag{2.17}$$

where η_2 is an unknown probability density function that has mean 0 given X. The case-control sample could be considered as a random sample from an imaginary infinite superpopulation, where the ratio between disease and normal is N_1/N_0. N_1 and N_0 here are group sizes in the case and control groups, respectively. The joint density of D, Y, X in the superpopulation is defined as:

$$f_{X,Y,D}(x, y, d) = \frac{N_d}{N} \frac{\eta_1(x)\eta_2(\epsilon, x)H(d, x, y, \alpha)}{p^{true}_D(d, \alpha, \beta, \eta_1, \eta_2)} \tag{2.18}$$

where $\theta = (\alpha^T, \beta^T)^T$ is the parameter of interest; $\eta_1(\cdot)$ and $\eta_2(\cdot, \cdot)$ are the nuisance parameters. An efficient estimator can be obtained by solving the semiparametric score equation

$$\sum_{i=1}^{N} [S(X_i, Y_i, D_i) - g\{Y_i - m(X_i, \beta)X_i\} - (1 - D_i)v_0 - D_i v_1] = 0 \tag{2.19}$$

where $S()$ is the score function and $g()$ is an arbitrary function. It is mentioned in the paper that the proposed estimator is not only efficient for the constructed superpopulation but also the real whole population.

2.3 Non-ignorbale Non-response

In this section, we review the existing approaches for missing data imputation and estimate parameters in the presence of non-response. We focus on the methods applicable to situations when the non-response is not missing at random (NMAR), that is to say the probability

a response is labelled depends on not only the observed but also the missing observations (Little and Rubin, 2019). In this case, the non-response cannot be ignored.

With the presence of non-ignorable non-response, disregarding such a missing mechanism may destroy the representativeness of the remaining samples and subsequently lead to significant estimation bias (Baker and Laird, 1988; Diggle and Kenward, 1994; Ibrahim et al., 1999; Molenberghs and Kenward, 2007). We assume that the problem of interest is to unbiasedly learn an outcome model $Y|\mathbf{x}$. Without loss of generality, when y is continuous, we consider a linear model given by

$$Y = \alpha + \mathbf{x}\beta + \epsilon, \tag{2.20}$$

where $\epsilon = (\epsilon_1, \cdots, \epsilon_N)^T \sim N(\mathbf{0}, \sigma^2 I)$ and $\epsilon \perp \mathbf{x}$ is the error term with zero unconditional mean, that is, $E(\epsilon_i) = 0$. We let $r_i \in \{0, 1\}$ be the "labeling indicator", where y_i is observed if and only if $r_i = 1$. With the non-ignorable missingness, dropping out missing data can lead to strongly biased estimates when r depends on y. The parameter estimates will not be consistent since $E\{\epsilon_i | r_i = 1\}$ and $E\{\epsilon_i x_i | r_i = 1\}$ are not zero. The missing values could not be imputed even if we would have consistent estimates since

$$E\{y_i | r_i = 0, x_i; \alpha, \beta\} = \frac{E\{y_i(1 - r_i)|x_i; \alpha, \beta\}}{1 - P(r_i = 1|x_i; \alpha, \beta)} = \alpha + \beta^T x_i - \frac{\text{cov}(y_i, \pi_i|x_i; \alpha, \beta)}{1 - E(\pi_i|x_i; \alpha, \beta)} \neq \alpha + \beta^T x_i. \tag{2.21}$$

Modeling non-ignorable missingness is challenging because the MNAR mechanism is usuallyunknown and may require additional model identifiability assumptions (Chen, 2001; Qin et al., 2002; Tang et al., 2003; Ibrahim et al., 2005). Little and Rubin (2019) classifies the approaches dealing with missing data into four different categories:

- **Methods based on completely observed units.** These approaches are completely based on the fully observed subjects while discarding observations that contain missing values. They are usually easy to implement in practice, but may result in estimation

inaccuracy because the complete cases are not randomly sampled from the whole population (Little and Rubin, 2019).

- **Weighting procedures.** These methods assign the inverse of estimated response probabilities as weights to the responding units (Robins et al., 1995; Carpenter et al., 2006) when building the likelihood function, but most of these procedures are designed for the missing at random (MAR) mechanism instead of NMAR. Very few methods, such as the one proposed by Deville (2000) and Chang and Kott (2008) can work for the non-ignorable non-response situation. The weighting approaches are usually based on the auxiliary information available for all the subjects, and the conditional probability to respond is always considered as propensity score (Rosenbaum and Rubin, 1983).

- **Imputation Procedure** Another class of methods is to impute missing data by using observed data (Rubin, 1976; Schafer and Schenker, 2000; Little and Rubin, 2019). These methods are based on the derived fully likelihood function including all the subjects, with non-respondents valued by estimations using information of respondents. The imputation procedures fall into two broad groups: single imputation and multiple imputation. Single imputation assigns a single value to each missing unit. The missing outcomes can be imputed by simply using the sample means or a random draw from the estimated conditional distribution (stochastic regression imputation). The disadvantage of single imputation is that it does not facilitate estimation of the variances due to non-response. To address this issue, multiple imputation can be employed by generating a set of plausible values for each missing unit based on several independent random draws from the posterior predictive distribution. The original multiple imputation method is proposed by Rubin (1976), and elaborated by Rubin (2004). The existing publications discussing imputation-based approaches include Glynn et al. (1993); Rubin (1996); Schafer (1997); Schafer and Schenker (2000).

- **Model-based procedures** These methods estimate the related parameters using the

likelihood function based on the fully observed units. The advantage of model-based methods is that they are flexible enough to handle both MAR and NMAR non-response. To account for the missingness of NMAR, model-based approaches are usually employed in two different ways: selection models or pattern-mixture models, which can be solved from the perspective of either Bayesian or frequentist.

Recently, two advanced methods have been proposed to facilitate model identification when dealing with non-ignorable missingness under the exponential tilting model (Kim and Yu, 2011). (Zhao et al., 2013; Tang et al., 2014) estimate the tilting model using external data, but such data is often unavailable in many applications, making these methods infeasible. The other method is to introduce an instrumental variable, which is associated with the response of interest but conditionally independent of the data missingness (Wang et al., 2014; Zhao and Shao, 2015; Yang et al., 2014; Shao and Wang, 2016).

In the rest of this section, we summarize the model-based approaches for non-ignorable non-response according to Sikov (2018). We assume that the covariate set x is observed for all the units and the response y is partially observed. We let $Y = (y_1, \ldots, y_r, y_{r+1}, \ldots, y_n) = (Y_{obs}; Y_{mis})$, $x = (x_1, \ldots, x_n)$ and $J = (R_1, \ldots, R_n)$. Specifically, Y_{obs} and Y_{mis} here represent the subsets of respondents and non-respondents, respectively. We can derive the pdf of the observed data as:

$$\begin{aligned} f(y_{obs}, J|x;\xi) &= f(y_1, \ldots, y_r, R_1, \ldots, R_n|x_1, \ldots, x_n, (1, \ldots, n) \in S; \xi) \\ &= \int \cdots \int f(y_1, \ldots, y_n, R_1, \ldots, R_n|x_1, \ldots, x_n, (1, \ldots, n) \in S; \xi) dy_{r+1}, \ldots dy_n \\ &= \prod_{i=1}^{r} f(y_i, R_i|x_i, i \in S; \xi) \prod_{i=r+1}^{n} \int f(y_i, R_i|x_i, i \in S; \xi) dy_i \end{aligned} \tag{2.22}$$

where ξ denotes the vector of unknown parameters related to the joint model. Both the two model-based methods could be non-identifiable unless some arbitrary modelling assumptions hold. More details about these two model settings will be discussed as follows.

2.3.1 Selection models

Under the framework of selection models, we have

$$f(y_i, R_i|x_i, i \in S; \xi = (\theta, \gamma)) = Pr(R_i|y_i, x_i, i \in S; \gamma) f_S(y_i|x_i; \theta) \tag{2.23}$$

where $f_S(y_i|x_i;\theta)$ and $Pr(R_i|y_i, x_i, i \in S;\gamma)$ model the sample pdf and missing mechanism, respectively. θ and γ are the parameters to estimate. In this case, the fully observed units can be seen as a sub-group, sampled in probabilities $Pr(R_i = 1|y_i, x_i, i \in S;\gamma)$. Based on the model specification, selection models works better when the main target of inference is the marginal distribution of the complete data. By assuming the sample outcomes are independent given the covariates, the fully likelihood can be written in the form:

$$\begin{aligned} L &= \int \dots \int \prod_{i=1}^{n} Pr(R_i = 1|y_i, x_i, i \in S;\gamma) f_S(y_i|x_i;\theta) dy_{r+1} \dots dy_n \\ &= \prod_{i=1}^{r} Pr(R_i = 1|y_i, x_i, i \in S;\gamma) f_S(y_i|x_i;\theta) \prod_{i=r+1}^{n} Pr(R_i = 0|x_i, i \in S;\theta,\gamma) \end{aligned} \tag{2.24}$$

where

$$Pr(R_i = 0|x_i, i \in S;\theta,\gamma) = 1 - \int Pr(R_i = 1|y_i, x_i, i \in S;\gamma) f_S(y_i|x_i;\theta) dy_i \tag{2.25}$$

The missing mechanism can be modelled as

$$Pr(R_i = 1|y_i, x_i, i \in S;\gamma) = g(\gamma_0 + x_i\gamma_1 + y_i\gamma_2) \tag{2.26}$$

with some function g valued in the range $(0, 1)$. In this case, the missing values can be imputed by the expectations $E_{R^c}(y_i|x_i) = E(y_i|x_i, R_i = 0)$ based on the Bayes theorem:

$$\begin{aligned} E_{R^c}(y_i|x_i) &= \int y_i f(y_i|x_i, i \in S, R_i = 0) dy_i \\ &= \frac{\int y_i P(R_i = 0|y_i, x_i, i \in S) f_S(y_i|x_i) dy_i}{\int P(R_i = 0|y_i, x_i, i \in S) f_S(y_i|x_I) dy_i} \end{aligned} \tag{2.27}$$

In practice, the probabilities and densities in (2.27) are replaced by the maximum likelihood estimations. The imputed values can also be obtained by drawing random samples from $f_{R^c}(y_i|x_i) = f(y_i|x_i, i \in S, R_i = 0)$. The frameworks of selection model are discussed in (Greenlees et al., 1982; Heckman, 1976; Ibrahim and Lipsitz, 1996; Peress, 2010). Selection model is able to estimate all the unknown parameters, but the use of the likelihood is inevitable based on strong distribution assumptions as noted by (Little, 1994).

Beaumont (2000) improves the model robustness by relaxing the normality assumption of the residuals. The parameter γ can be estimated by maximizing the response likelihood:

$$L = \prod_{i=1}^{r} Pr(R_i = 1|y_i, x_i, i \in S; \gamma) \prod_{i=r+1}^{n} Pr(R_i = 0|x_i, i \in S; \theta, \gamma)$$

with respect to γ, assuming that θ is known. Similarly, estimation of θ can be obtained by solving a weighted least square equations, given γ. The estimation procedure is updated iteratively until convergence. Specifically, they expand $Pr(R_i = 1|y_i, x_i; i \in S; \gamma)$ around the mean $E_S(y_i|x_i) = \beta^t x_i$. The imputed missing outcomes obtained by the expectations with respect to the sample distribution $\hat{E}_S(y_i|x_i) = E_S(y_i|x_i; \hat{\theta}, \hat{\gamma})$ is biased since the the missing outcomes must be imputed either by $\hat{E}_{R^c}(y_i|x_i)$ or by random sample drawn from the distribution $f_{R^c}(y_i|x_i; \hat{\theta}, \hat{\gamma})$.

Overall, selection models are more intuitive to implement in practice but the modelling of NMAR may be non-identifiable and thus require unverifiable model assumptions.

2.3.2 Pattern-mixture models

Different from selection models, pattern-mixture models formulate distinct models for response and non-response units:

$$f(y_i, R_i|x_i, i \in S; \xi = (\psi^{(l)}, \psi_r)) = f(y_i|x_i; \psi_m^{(l)}) Pr(R_i|x_i, i \in S; \psi_r) \tag{2.28}$$

where $f(y_i|x_i; \psi_m^{(l)}, l = 0, 1)$ and $Pr(R_i|x_i, i \in S; \psi_r)$ model the pdf of Y under the different patterns of the missing data and the response probability given sample selection, respectively. $l = 1$ corresponds to the respondents and $l = 0$ for the non-respondents. In this case, the likelihood function can be defined as:

$$\begin{aligned} L &= \int \dots \int \prod_{i=1}^{n} f(y_i|x_i; \psi_m^{(l)}) Pr(R_i|x_i, i \in S; \psi_r) dy_{r+1} \dots dy_n \qquad (2.29) \\ &= \prod_{i=1}^{r} f_S(y_i|x_i, i \in S; \psi_m^{(l)}) Pr(R_i = 1|x_i, i \in S; \psi_r) \prod_{i=r+1}^{n} Pr(R_i = 0|x_i, i \in S; \psi_r) \end{aligned}$$

Similar to the selection models, the unverifiable assumptions is necessary to obtain the identification. Specifically, the factorization (2.28) partitions the parameters of full-data model into the identified and non-identified sets. The parameters related to the respondents' model $f(y_i|x_i; \psi_m^{(1)})$ and the probability to respond $Pr(R_i|y_i, x_i, i \in S; \psi_r)$ can be identified. The parameters corresponding to the non-respondent model $f(y_i|x_i; \psi_m^{(0)})$ are not identifiable from the data. Identification of the pattern-mixture models is based on the postulating unverifiable links among the distributions of the outcomes conditional on the patterns of non-response. Little (1994) explores the potential relationships between the parameters governing the models holding for different missingness patterns, and compare pattern-mixture and selection models by some real examples. Chambers et al. (2012) studies the applications of pattern-mixture models in the situation when some non-respondents are available through a more intensive follow-up survey.

Different from the selection models, pattern-mixture models split the whole parameter

set into the identified and un-identified parts, and build a framework for sensitivity analysis (Thijs et al., 2002; Daniels and Hogan, 2008). The weakness of pattern-mixture models is that the model for non-responding units $f(y_i|x_i;\psi_m^{(0)})$ can not be obtained from the fitted models $f(y_i|x_i;\psi_m^{(1)})$ and $Pr(R_i|y_i,x_i,i\in S;\psi_r)$. Moreover, the parameters associated with the distribution for the complete respondents can not be easily estimated, which requires marginalization of the distribution of outcomes over non-response patterns.

2.3.3 Pseudo-likelihood method

Tang et al. (2003) proposes a 'pseudo-likelihood' method using the conditional pdf $f_S(x_i|y_i)$ for the responding units, where the specification of this sample pdf and the marginal pdf $g_S(x_i)$ is required. The method assumes that the probability to respond only depends on y, i.e. $g_R(x_i|y_i) = g_S(x_i|y_i)$, where $g_R(x_i|y_i)$ is the conditional pdf for a respondent. The likelihood is defined as

$$L = \prod_{i=1}^{r} g_S(x_i|y_i;\theta,\eta) = \prod_{i=1}^{r} \frac{f_S(y_i|x_i;\theta)g_S(x_i;\eta)}{\int f_S(y_i|x_i;\theta)g_S(x_i;\eta)dx_i} \tag{2.30}$$

Although the product only covers the responding units, estimations of $g_S(x_i)$ requires the covariates to be known for all the observations. The method combines the estimation of $g_S(x_i;\eta)$ based on the complete units with the conditional distribution $f_S(x_i|y_i;\theta)$ using the fully observed units. They propose a two-step procedure to estimate θ and η:

1. Estimate η as $\hat{\eta} = \arg\max_\eta \prod_{i=1}^{n} g_S(x_i;\eta)$ or as $\hat{\eta} = G_n(x)$, where $G_n(x)$ is the empirical sample distribution of X

2. Estimate θ by maximizing the likelihood (2.30) with η replaced by $\hat{\eta}$.

Although they demonstrates that this method is robust to the mis-specification of the missing mechanism, it is less efficient than selection models when the responding probability is correctly specified. They discuss the case when the responding probability depends on $Y^* = Y + \lambda^t X$. If λ is known, the pseudo-likelihood method can be applied to data (X, Y^*).

2.4 Spatial-Temporal Predictions

Data-driven prediction for spatial-temporal traffic systems has drawn wide attention for decades. The main target of these problems is to predict the expected value at each spatial location within an incoming time window based on the system dynamics learned from historical data. In this section, we discuss some state-of-the-art methods for spatial-temporal traffic predictions and their limitations when applied to origin-destination flow data.

A large number of approaches have been proposed for spatial-temporal prediction problems, most of which fall into two main groups: traditional statistical methods and more advanced deep learning methods. Some early statistical methods including Auto-regressive integrated moving average (ARIMA), Kalman filtering, and their variants, model the spatial-temporal data as multi-dimensional time-series, which cannot capture enough spatial information (Li et al., 2012; Lippi et al., 2013; Moreira-Matias et al., 2013; Shekhar and Williams, 2008). Idé and Sugiyama (2011); Zheng and Ni (2013) smooth the spatial similarities among nearby locations based on the road networks and time sequences according to given regularizations. Kwon and Murphy (2000); Yang et al. (2013) capture the spatial-temporal correlations by using Hidden Markov Model, which can only work for small-scale traffic data. However, all these approaches use some pre-calculated spatial features instead of capturing the correlations among different OD flows by the model itself when predicting future OD flow values. Deng et al. (2016) learns the time-dependent latent attributes by finding the optimal decomposition of the dynamic traffic flow matrices. Their method assumes that the latent attribute representations constantly evolve with time. However, some recurring incidents or emergency situations can result in non-stationarity.

Deep learning enables prediction models to automatically extract non-linear spatial patterns inside the OD flow data. Wei et al. (2016) introduces a Zero-Grid Ensemble Spatio Temporal model (ZEST), which integrates a temporal predictor and a spatial predictor through a fully connected network for the final prediction. Wang et al. (2017) presents an end-to-end framework, called Deep Supply-Demand (DeepSD), which utilizes multiple data

sources to improve the prediction performance. All theses methods model the spatial and temporal representations, respectively, without building a dynamic connection.

To address this issue of dynamic connection, some recent studies use convolutional LSTM to jointly capture the spatial-temporal dependency. Zhang et al. (2016, 2017) model the city as an image by dividing the whole area into small grids and employed residual neural network to capture the temporal closeness, period, and trend properties of traffic flows. Ma et al. (2017) applies CNN to the image built on the whole city area. Another set of studies utilize recurrent-neural-network to model the temporal sequential correlations. Yu et al. (2017) proposes an end-to-end deep Long-short-term memory (LSTM) model to forecast peak-hour and post-accident traffic situation. Cui et al. (2016) introduces an unidirectional LSTM (SBU-LSTM) neural network, which considers both forward and backward dependencies of time sequences for traffic speed prediction. All the methods discussed above explicitly model spatial and temporal dependencies respectively, but still can not build the connections between the both sides. To address this issue, some recent studies try convolutional LSTM to model the spatial-temporal dependency (Xingjian et al., 2015; Ke et al., 2017; Zhou et al., 2018).

Yao et al. (2018) introduces a mult-view spatial-temporal prediction model, consisting of both spatial and temporal views to jointly obtain the spatial-temporal relations. The goal of the paper is to predict taxi demand at each local region within the incoming predicting time window give the historical information.

At each time interval t, Yao et al. (2018) treats one spot i with its surrounding neighborhood as an $S \times S$ image with one channel including the grid-level demand amount, denoted by $Y_t^i \in R^{S \times S \times 1}$. For the spatial-view, a zero-padding local CNN operation takes Y_t^i as the input $Y_t^{i,0}$ and feeds it into K layers, where the transformation at k-th layer is defines as:

$$Y_t^{i,k} = f(Y_t^{i,k-1} * W_t^k + b_t^k) \tag{2.31}$$

where $*$ denotes the convolutional operation and $f(x)$ is the ReLu function $\max(x, 0)$. The output representations $Y_t^{i,k} \in R^{S \times S \times \lambda}$ after K convolution layers is flattened into a feature vector $s_t^i \in R^{S^2\lambda}$. Then a fully connected layer reduce the dimension of s_t^i from $S^2\lambda$ to d by

$$\hat{s}_i^t = f(W_t^{fc} s_t^i + b_t^{fc}) \tag{2.32}$$

The spatial features obtained by local CNN at time t is then concatenated with some external context features e_i^t to get

$$g_i^t = s_i^t \oplus e_i^t \tag{2.33}$$

g_i^t is then fed into a LSTM model to learn the sequential correlations in temporal dimension:

$$h_i^t = \text{LSTM}(h_i^{t-1}, g_i^t) \tag{2.34}$$

to make the output of LSTM h_i^t contains both temporal and spatial information. h_i^t is concatenated with the global-view features m_i^t obtained through network embedding to get the input q_i^t for the final prediction layer, which is defined as:

$$\hat{y}_{t+1}^i = \sigma(W_f q_i^t + b_f) \tag{2.35}$$

where W_f and b_f are learnable parameters. $\sigma(x)$ is a Sigmoid function to gurantee the value range of predictions within $[0, 1]$ as the real demand values are normalized for better prediction performance. Cheng et al. (2018) also combines CNN and RNN together to obtain spatial-temporal correlations, while the difference is that it applies CNN to the whole image instead of using local CNN as Yao et al. (2018) did. Yao et al. (2018) improves Yao et al. (2018)'s method by designing a periodically shifted attention mechanism to capture the long-term periodic influence and temporal shifting in time series prediction. Moreover, they proposed a flow gating mechanism to learn the location similarities by incorporating the directed traffic flows other than only using the non-directed demand value. When applied to

OD predictions, most of these CNN-based methods treat each snapshot $O_t \in R^{N \times N}$ including all the N^2 OD flows as an image. In this case, some nearby OD flows in O_t covered by a single CNN kernel may not be semantically correlated. On the other hand, two neighboring OD flows with shared vertexes in the graph can be far from each other in terms of images.

As we mentioned above, many real-world datasets have graph structures, including social networks, knowledge graphs or some large-scale spatial-temporal traffic systems. The traditional Convolution Neural Network (CNN) can not be directly applied since CNN can only capture the spatial information from the perspective of images. However, some graph vertexes far away from each in the image space may be topologically close and semantically correlated.

Seo et al. (2018) combines the graph convolutional networks (denoted by CNN_G) to identify spatial structures with recurrent neural network (RNN) to find dynamic patterns. Two different appraoches have been discussed. The first is to use GCN to extract spatial representations at each time t as the input for the LSTM model:

$$\begin{aligned} x_t^{CNN} &= CNN_G(x_t) \\ i_t &= \sigma(W_{xi}x_t^{CNN} + W_{hi}h_{t-1} + w_{ci} \odot c_{t-1} + b_i), \\ f_t &= \sigma(W_{xf}x_t^{CNN} + W_{hf}h_{t-1} + w_{cf} \odot c_{t-1} + b_f), \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t^{CNN} + W_{hc}h_{t-1} + b_c), \\ o_t &= \sigma(W_{xo}x_t^{CNN} + W_{ho}h_{t-1} + w_{co} \odot c_t + b_o), \\ h_t &= o_t \odot \tanh(c_t). \end{aligned} \tag{2.36}$$

where $x_t \in R^{n \times d_x}$ is input matrix. The other replaces the Euclidean 2D convolution by graph

convolution in convLSTM model proposed by Xingjian et al. (2015):

$$\begin{aligned} i &= \sigma(W_{xi} *_G x_t + W_{hi} *_G h_{t-1} + w_{ci} \odot c_{t-1} + b_i), \\ f &= \sigma(W_{xf} *_G x_t + W_{hf} *_G h_{t-1} + w_{cf} \odot c_{t-1} + b_f), \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc} *_G x_t + W_{hc} *_G h_{t-1} + b_c), \\ o_t &= \sigma(W_{xo} *_G x_t + W_{ho} *_G h_{t-1} + w_{co} \odot c_t + b_o), \\ h_t &= o_t \odot \tanh(c_t). \end{aligned} \tag{2.37}$$

where $W_{xi} *_G x_t$ represents the graph convolution of x_t with $d_h d_x$ filters which are functions of the graph Laplacian L parametrized by K Chebyshev coefficients.

Yan et al. (2018) proposes a novel model in dynamic skeletons called Spatial-Temporal Graph Convolutional Networks (ST-GCN) by applying the graph CNN to the spatial-temporal domain to jointly learn the spatial and temporal features. Specifically, Yan et al. (2018) extends the concept of neighborhood to also include temporally connected nodes. Manessi et al. (2017) also uses the idea to combine LSTM and GCN in semi-supervised classification problems, where graphs are allowed to be dynamic with structures changing during time. However, none of them are directly applicable to the prediction problem of OD flow data since both the input and output of GCNs are node-level features. For OD flow prediction problems, the spatial information in edge space is more important because of the equivalence between OD flows and graph edges by our definition.

CHAPTER 3: ANALYSIS OF SECONDARY PHENOTYPES IN MULTI-GROUP ASSOCIATION STUDIES

3.1 Introduction

To motivate the proposed methodology, we consider a large database with imaging, genetic, and clinical data from 1737 subjects collected through the Alzheimer's Disease Neuroimaging Initiative (ADNI) (http://www.adni-info.org/). The overall design of the ADNI is a longitudinal study of various biomarkers at baseline and their longitudinal profiles. ADNI has gone though four phases from ADNI1, GO, 2 to ADNI3 from 2004 until 2016. ADNI1 began with 204 cognitively normal controls (NC), 362 subjects with mild cognitive impairment (MCI), and 179 subjects with Alzheimer's disease (AD), and was extended by three follow-up phases with different number of subjects in each category. ADNI is a typical example of multi-group studies. Similar to the case-control design, the multi-group sample is usually not a random sample from the whole population because of the unequal selection probabilities between different disease groups. The proportions of AD and MCI in ADNI are much bigger than their global prevalences in the age-matched general population (Kim et al., 2015). In this paper, we focus on the brain regions of the left and right hippocampi of each ADNI subject and a large genetic data set with over 6,000,000 genotyped and imputed single-nucleotide polymorphisms (SNPs) on all 22 human chromosomes. Since the hippocampus is critical for learning and memory and is vulnerable to damage in the early stages of AD (Schuff et al., 2009), the volume and shape of the hippocampi may be effective phenotypes that facilitate the identification of causal genes and the mechanistic understanding of pathophysiological processes of AD. Our primary goal is to search for genetic patterns that are associated with local hippocampal changes, while correcting for the selection bias associated with ascertainment in multi-group studies.

In many genetic association studies, some variables of interest are the marker genotype(s), G, secondary (or intermediate) traits Y, the primary phenotype (multi-group status) D, clinical variables $\boldsymbol{C}$, and the ascertainment (sampling) indicator S. For instance, various imaging measures (e.g., subcortical volumes) have been widely used as secondary traits that may be directly associated with a specific disease outcome for most brain-related diseases. A statistical challenge arises from the fact that the main target of interest is the population model of Y given G, whereas both secondary traits Y and marker genotype(s) G are collected conditional on the grouping phenotype D. In genetic epidemiology, standard statistical methods that either ignore ascertainment or naively adjust for ascertainment by conditioning on the disease status (e.g., meta-analysis of subjects in different subgroups) can lead to estimation bias, an inflated false-positive rate, and decreased statistical power. Therefore, it may be critical to adjust for D when one models Y given G in these genetic association studies.

There is a large literature on the development of statistical methods for eliminating the selection bias associated with ascertainment in case-control (or two-group) studies. The simplest method is to fit a regression model to all subjects in a single group (e.g., cases or controls, or each subgroup in multi-group study). It requires a strong assumption that no group difference exists in the genetic effects regarding the corresponding secondary traits. Moreover, dropping a certain number of observations can substantially decrease the estimation efficiency and statistical power. Another simple method, called LRegD (Potkin et al., 2010), is to include the case-control status D as an additional covariate in the regression models. However, LRegD may yield flawed conclusions, since the associations between a secondary outcome and an exposure of interest in the case and control groups can be quite different from that in the underlying target population (Tchetgen Tchetgen, 2014). Various weighted likelihoods, such as the inverse probability weighting (IPW) approach, have been widely used (Richardson et al., 2007; Monsees et al., 2009; Schifano et al., 2013; Sofer et al., 2017), but they do not utilize the information collected on the primary outcome D. Lee et al. (1997)

and Jiang et al. (2006) develop a maximum likelihood estimate of the regression coefficients assuming that the sampling rates for cases and controls are known. Lin and Zeng (2009) introduces a retrospective likelihood function by explicitly conditioning on the sampling scheme. He et al. (2012) uses a Gaussian copula approach, allowing more flexible distributions of the secondary outcome Y compared to Lin and Zeng (2009). Wei et al. (2013) proposes a robust estimation method for secondary analysis of case-control data by assuming that the secondary trait Y follows a homoscedastic regression model given X. Breslow et al. (2000) applies the semiparametric inference method through building an augmented estimation equation to improve the efficiency of IPW. Song et al. (2016) introduces a set of counterfactual estimation functions under an alternative disease status and combines the observed and counterfactual estimation functions into a set of weighted estimation equations. However, all these approaches focus on the case-control design.

Our aim is to develop a general regression framework for the analysis of secondary phenotypes collected in multi-group association studies, called MGLREG. There are two major contributions.

(I) To the best of our knowledge, we are the first that systematically discusses the secondary trait analysis in multi-group studies such as ADNI, while allowing the multiple-phase design.

(II) We have developed companion software, called MGLREG, along with its documentation and released it to the public through https://github.com/BIG-S2/MGLREG.

3.2 Methods

In Section 3.2.1, we introduce the data structure and some notations. In Sections 3.2.2 and 3.2.3, we build the conditional model for Y given D and $\boldsymbol{X}$ and derive its associated estimation equations for the three-group study, that is, $J = 3$. Our approach can be easily extended from the basic $J = 3$ case to the more general setting of $J > 3$ (details for general J discussed in supplements). In Section 3.2.4, we discuss how to extend our regression framework from continuous secondary outcomes to binary ones. In Section 3.2.5, we further consider the extension to multiple phases scenario.

3.2.1 Data Structure and Notation

Suppose that we consider N independent subjects from a multi-group study. For each subject, given the group status $D_i \in \{0, 1, \ldots, J-1\}$, we denote S_i as the ascertainment (sampling) indicator and observe the secondary phenotype Y_i of interest, the clinical factors $\boldsymbol{C}_i$, as well as the genotype score G_i for $i = 1, \ldots, N$, where J is a positive integer. For instance, $J = 2$ corresponds to the case-control design, whereas $J > 2$ corresponds to the multi-group design. Without loss of generality, we focus on continuous secondary traits, while the group 0 corresponds to the control group. Suppose there are n_j subjects in the $j-$th group for $j = 0, \ldots, J-1$ such that N is equal to $n_0 + n_1 + \ldots + n_{J-1}$. An important assumption is that the prevalence of each subgroup j is known to be $\widetilde{p}_j = P(D = j)$ in the target population and $\widetilde{\pi}_j = P(D = j|S = 1) = n_j/N$ in the sample for $j = 0, 1, \ldots, J-1$. Although the true value of $\widetilde{p}_j$ is required, our method still works for an approximated value of $\widetilde{p}_j$. To demonstrate this point, we allow misspecification of $\tilde{p}_j$ in the simulation studies and find that our method performs acceptably stable with varied $\tilde{p}_j$'s combinations.

3.2.2 Model Setup

The main target of inference is the population mean model for Y given $\mathbf{X}$, denoted as $\mu(\boldsymbol{X}) = E(Y|\boldsymbol{X})$. We focus on the three-group case with $J = 3$ from now on, but all derivations given below are valid when we replace 2 by $J-1$. By using the law of conditional expectations, we have

$$\mu(\boldsymbol{X}) = \sum_{j=0}^{2} \widetilde{\mu}(\boldsymbol{X}, D = j) \times P(D = j|\boldsymbol{X}), \tag{3.1}$$

where $\widetilde{\mu}(\boldsymbol{X}, D) = E(Y|\boldsymbol{X}, D)$. A sufficient condition for estimating $\mu(\boldsymbol{X})$ is to estimate both $\widetilde{\mu}(\boldsymbol{X}, D)$ and $P(D|\boldsymbol{X})$. Since we observe Y and $\boldsymbol{X}$ conditional on D and $S = 1$, we can consistently estimate $E(Y|\boldsymbol{X}, D, S = 1)$ and $P(D|\boldsymbol{X}, S = 1)$ instead of $\widetilde{\mu}(\boldsymbol{X}, D)$ and $P(D|\boldsymbol{X})$.

The sampling design of the multi-group study depends on D only and therefore $(Y, \boldsymbol{X})$ is

randomly sampled within each group D. Accordingly, we could characterize a relationship between $E(Y|\boldsymbol{X}, D, S = 1)$ and $\widetilde{\mu}(\boldsymbol{X}, D)$ as:

$$\widetilde{\mu}(\boldsymbol{X}, D) = E(Y|\boldsymbol{X}, D) = E(Y|\boldsymbol{X}, D, S = 1). \tag{3.2}$$

It then follows from (3.2) that $\widetilde{\mu}(\boldsymbol{X}, D)$ can be consistently estimated.

Second, we characterize a relationship between $P(D|\boldsymbol{X}, S = 1)$ and $P(D|\boldsymbol{X})$. Let $\Pi_j(\boldsymbol{X}) = P(D = j|\boldsymbol{X}, S = 1)$ denote the risk function of $D = j$ at $\boldsymbol{X}$ in the multi-group sample and $P_j(\boldsymbol{X}) = P(D = j|\boldsymbol{X})$ be the probability of D given $\boldsymbol{X}$ in the whole population. For each $j = 0, 1, 2$, $\Pi_j(\boldsymbol{X})$ and $P_j(\boldsymbol{X})$ satisfy the following relationship:

$$\frac{\Pi_j(\boldsymbol{X})}{\Pi_0(\boldsymbol{X})} \cdot \frac{\widetilde{\pi}_0}{\widetilde{\pi}_j} = \frac{P_j(\boldsymbol{X})}{P_0(\boldsymbol{X})} \cdot \frac{\widetilde{p}_0}{\widetilde{p}_j}. \tag{3.3}$$

We assume that $\Pi_j(\boldsymbol{X})$ follows a multinomial logistic regression model as follows:

$$\log\left\{\frac{\Pi_j(\boldsymbol{X})}{\Pi_0(\boldsymbol{X})}\right\} \;=\; \log\left\{\frac{P_j(\boldsymbol{X})}{P_0(\boldsymbol{X})}\right\} + \eta_j = \boldsymbol{X}^T\boldsymbol{\varphi_j} \tag{3.4}$$

for $j = 0, 1$, and 2, where $\eta_j = \log(\widetilde{p}_0\widetilde{\pi}_j) - \log(\widetilde{p}_j\widetilde{\pi}_0)$. If the η_js are known and the ratio of $\Pi_j(\boldsymbol{X})$ over $\Pi_0(\boldsymbol{X})$ can be consistently estimated, then the ratio of $P_j(\boldsymbol{X})$ over $P_0(\boldsymbol{X})$ can be consistently estimated.

We derive a conditional model of $\widetilde{\mu}(\boldsymbol{X}, D)$ based on (3.2). Specifically, it follows from the equality $\sum_{j=0}^{2} P(D = j|\boldsymbol{X}) = 1$ and (3.2) that $\widetilde{\mu}(\boldsymbol{X}, j)$ is given by

$$\widetilde{\mu}(\boldsymbol{X}, j) = \mu(\boldsymbol{X}) + \sum_{k \neq j} P(D = k|\boldsymbol{X})\{\widetilde{\mu}(\boldsymbol{X}, j) - \widetilde{\mu}(\boldsymbol{X}, k)\}. \tag{3.5}$$

Furthermore, we define $\gamma_1(\boldsymbol{X}) = \widetilde{\mu}(\boldsymbol{X}, 1) - \widetilde{\mu}(\boldsymbol{X}, 0)$ and $\gamma_2(\boldsymbol{X}) = \widetilde{\mu}(\boldsymbol{X}, 2) - \widetilde{\mu}(\boldsymbol{X}, 0)$. With

some algebraic calculations, we can rewrite (3.5) as follows:

$$\widetilde{\mu}(\boldsymbol{X}, j) = \mu(\boldsymbol{X}) + \sum_{k=1}^{2} \{1(j = k) - P(D = k|\boldsymbol{X})\}\gamma_k(\boldsymbol{X}) \tag{3.6}$$

for $j = 0, 1$, and 2. The term besides $\mu(\boldsymbol{X})$ on the right-hand side of (3.6) encodes the selection bias by modeling the group difference of Y given different D statuses with fixed $\boldsymbol{X}$ (Tchetgen Tchetgen, 2014).

Equation (3.6) has several important implications. If the selection bias is absent, then we have $\gamma_1(\boldsymbol{X}) = \gamma_2(\boldsymbol{X}) = 0$ and $\widetilde{\mu}(\boldsymbol{X}, i)$ reduces to $\mu(\boldsymbol{X})$ regardless of the status of D. If the disease is rare, then both $P(D = 1|\boldsymbol{X})$ and $P(D = 2|\boldsymbol{X})$ are close to zero in the whole population and (3.6) reduces to

$$\widetilde{\mu}(\boldsymbol{X}, j) = \mu(\boldsymbol{X}) + \sum_{k=1}^{2} 1(j = k) \times \gamma_k(\boldsymbol{X}). \tag{3.7}$$

Furthermore, if we set $\gamma_1(\boldsymbol{X}) = \boldsymbol{X}^T\boldsymbol{\Gamma}_1$, $\gamma_2(\boldsymbol{X}) = \boldsymbol{X}^T\boldsymbol{\Gamma}_2$, and $\mu(\boldsymbol{X}) = \boldsymbol{X}^T\boldsymbol{\beta}$, where $\boldsymbol{\Gamma}_1$, $\boldsymbol{\Gamma}_2$, and $\boldsymbol{\beta}$ are three vectors of regression coefficients, then model (3.7) reduces to

$$\widetilde{\mu}(\boldsymbol{X}, j) = \boldsymbol{X}^T\boldsymbol{\beta} + \sum_{k=1}^{2} 1(j = k)\boldsymbol{X}^T\boldsymbol{\Gamma}_k, \tag{3.8}$$

in which $\boldsymbol{\beta}$ represents the main effects of $\boldsymbol{X}$ on Y and $\boldsymbol{\Gamma}_1$ and $\boldsymbol{\Gamma}_2$ represent the interaction effects of D and $\boldsymbol{X}$ on Y. However, if the disease is not rare, then the selection bias can be substantial when $\widetilde{\mu}(\boldsymbol{X}, D)$ varies dramatically across D.

3.2.3 Estimation

Our conditional model consists of three key components including (3.2), (3.4), and (3.6). We can develop a two-stage estimation procedure to estimate the parameters of interest in $\mu(\boldsymbol{X})$, $\{\gamma_j(\boldsymbol{X}) : j = 1, 2\}$ and $\{P_j(\boldsymbol{X}) : j = 1, 2\}$ as follows.

- Stage I: Based on (3.4), we can construct a set of estimation equations to estimate the unknown parameters in $P_j(\boldsymbol{X})$ in order to obtain its estimate, denoted as $\widehat{P}_j(\boldsymbol{X})$.

- Stage II: We can substitute $\widehat{P}_j(\boldsymbol{X})$ in (3.6) and then construct the other set of estimation equations to estimate the parameters in $\mu(\boldsymbol{X})$, $\gamma_1(\boldsymbol{X})$, and $\gamma_2(\boldsymbol{X})$ based on (3.6).

In Stage I, we assume that $\log\{P_j(\boldsymbol{X})\} - \log\{P_0(\boldsymbol{X})\} = f_1(\boldsymbol{X}; \boldsymbol{\varphi}_j, \eta_j)$ holds for $j = 1, 2$, where $f_j(\cdot;\cdot,\cdot)$ is a known parametric function. For instance, in (3.4), we set $f_1(\boldsymbol{X}; \boldsymbol{\varphi}_j, \eta_j) = \boldsymbol{X}^T\boldsymbol{\varphi}_j - \eta_j$ for each j. Since $\eta_j = \log(\widetilde{p}_0\widetilde{\pi}_j) - \log(\widetilde{p}_j\widetilde{\pi}_0)$ is known, we can construct a log pseudo-likelihood function, denoted as $L(\boldsymbol{\varphi})$, to estimate unknown parameters $\boldsymbol{\varphi} = (\boldsymbol{\varphi}_1^T, \boldsymbol{\varphi}_2^T)^T$ in $\{\Pi_j(\boldsymbol{X})\}$ based on N observations in the sample $\{(\boldsymbol{X}_i, D_i, S_i = 1) : i = 1, \ldots, N\}$. Specifically, the log pseudo-likelihood function $L(\boldsymbol{\varphi})$ is given by

$$\sum_{i=1}^{N}\left[\sum_{j=1}^{2}\{1(D_i = j)\boldsymbol{X}_i^T\boldsymbol{\varphi}_j\} - \log\{1 + \sum_{j=1}^{2}\exp(\boldsymbol{X}_i^T\boldsymbol{\varphi}_j)\}\right]. \tag{3.9}$$

We can calculate the maximum pseudo-likelihood estimate, $\widehat{\boldsymbol{\varphi}} = (\widehat{\boldsymbol{\varphi}}_{\mathbf{1}}^T, \widehat{\boldsymbol{\varphi}}_{\mathbf{2}}^T)^T = \text{argmax}_{\boldsymbol{\varphi}}\, L(\boldsymbol{\varphi})$ or equivalently, $\partial L(\widehat{\boldsymbol{\varphi}})/\partial\boldsymbol{\varphi}^T = \mathbf{0}$. Then, we compute

$$\widehat{P}_j(\boldsymbol{X}) = \exp\{f_j(\boldsymbol{X}; \widehat{\boldsymbol{\varphi}}_{\boldsymbol{j}}, \eta_j)\}/[1 + \exp\{f_1(\boldsymbol{X}; \widehat{\boldsymbol{\varphi}}_{\mathbf{1}}, \eta_1)\} + \exp\{f_2(\boldsymbol{X}; \widehat{\boldsymbol{\varphi}}_{\mathbf{2}}, \eta_2)\}]$$

as a consistent estimate of $P_j(\boldsymbol{X})$ for $j = 1$ and 2.

In Stage II, we need to assume an explicit form of $\mu(\boldsymbol{X})$, $\gamma_1(\boldsymbol{X})$, and $\gamma_2(\boldsymbol{X})$ as follows:

$$\mu(\boldsymbol{X}) = \mu(\boldsymbol{X}; \boldsymbol{\beta}), \quad \gamma_1(\boldsymbol{X}) = g_1(\boldsymbol{X}; \boldsymbol{\Gamma}_1), \quad \text{and} \quad \gamma_1(\boldsymbol{X}) = g_2(\boldsymbol{X}; \boldsymbol{\Gamma}_2), \tag{3.10}$$

where $\mu(\cdot,\cdot)$, $g_1(\cdot,\cdot)$, and $g_2(\cdot,\cdot)$ are known functions and $\boldsymbol{\beta}$, $\boldsymbol{\Gamma}_1$, and $\boldsymbol{\Gamma}_2$ are unknown parameter vectors. Suppose that $\boldsymbol{\theta} = (\boldsymbol{\beta}^T, \boldsymbol{\Gamma}_1^T, \boldsymbol{\Gamma}_2^T)^T$ and $\mu(\cdot,\cdot)$, $g_1(\cdot,\cdot)$, and $g_2(\cdot,\cdot)$ are all in the linear form as described in last section. In this case, (3.6) can be rewritten as

$$\widetilde{\mu}(\boldsymbol{X}, D; \boldsymbol{\theta}, \widehat{\boldsymbol{\varphi}}) = \mu(\boldsymbol{X}; \boldsymbol{\beta}) + \sum_{j=1}^{2}\{1(D = j) - \widehat{P}_j(\boldsymbol{X}; \widehat{\boldsymbol{\varphi}})\}g_j(\boldsymbol{X}; \boldsymbol{\Gamma}_j). \tag{3.11}$$

We construct consistent estimation equations based on N observations $\{(y_i, \boldsymbol{X}_i, D_i, S_i = 1) :$

$i = 1, \ldots, N\}$ as follows:

$$U(\boldsymbol{\theta}; \widehat{\boldsymbol{\varphi}}) = \sum_{i=1}^{N} \frac{\partial \widetilde{\mu}(\boldsymbol{X}_i, D_i; \boldsymbol{\theta}, \widehat{\boldsymbol{\varphi}})}{\partial \boldsymbol{\theta}^T} \epsilon_i(\boldsymbol{\theta}, \widehat{\boldsymbol{\varphi}}) = \mathbf{0}, \tag{3.12}$$

where $\epsilon_i(\boldsymbol{\theta}, \widehat{\boldsymbol{\varphi}}) = y_i - \widetilde{\mu}(\boldsymbol{X}_i, D_i; \boldsymbol{\theta}, \widehat{\boldsymbol{\varphi}})$ for $i = 1, \ldots, N$. Let $\widehat{\boldsymbol{\theta}}$ be the solution to $U(\boldsymbol{\theta}; \widehat{\boldsymbol{\varphi}}) = \mathbf{0}$ such that $U(\widehat{\boldsymbol{\theta}}; \widehat{\boldsymbol{\varphi}}) = \mathbf{0}$.

The algorithm which jointly solves $U(\widehat{\boldsymbol{\theta}}; \widehat{\boldsymbol{\varphi}}) = \mathbf{0}$ and $\partial L(\widehat{\boldsymbol{\varphi}})/\partial \boldsymbol{\varphi}^T = \mathbf{0}$ is denoted as "MGLReg" throughout the chapter. We can show that

$$\sqrt{n} \begin{pmatrix} \widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_* \\ \widehat{\boldsymbol{\varphi}} - \boldsymbol{\varphi}_* \end{pmatrix} \rightarrow^L N(\mathbf{0}, \boldsymbol{\Sigma}), \tag{3.13}$$

where $\rightarrow^L$ denotes the convergence in distribution and $\boldsymbol{\theta}_*$ and $\boldsymbol{\varphi}_*$ are the true value of $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$, respectively. Moreover, $\boldsymbol{\Sigma}$ as a covariance matrix can be approximated by $\widehat{\boldsymbol{\Sigma}}$, which is given by

$$\begin{pmatrix} \frac{1}{N}\partial_{\boldsymbol{\theta}} U(\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\varphi}}) & \frac{1}{N}\partial_{\boldsymbol{\varphi}} U(\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\varphi}}) \\ \mathbf{0} & \frac{1}{N}\partial_{\boldsymbol{\varphi}^2} L(\widehat{\boldsymbol{\varphi}}) \end{pmatrix}^{-1} \widehat{\mathrm{Cov}} \begin{pmatrix} \frac{U(\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\varphi}})}{\sqrt{N}} \\ \frac{\partial_{\boldsymbol{\varphi}} L(\widehat{\boldsymbol{\varphi}})}{\sqrt{N}} \end{pmatrix} \begin{pmatrix} \frac{1}{N}\partial_{\boldsymbol{\theta}} U(\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\varphi}}) & \frac{1}{N}\partial_{\boldsymbol{\varphi}} U(\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\varphi}}) \\ \mathbf{0} & \frac{1}{N}\partial^2_{\boldsymbol{\varphi}} L(\widehat{\boldsymbol{\varphi}}) \end{pmatrix}^{-T}, \tag{3.14}$$

where $\partial_{\boldsymbol{\theta}} = \partial/\partial\boldsymbol{\theta}$ and $\partial_{\boldsymbol{\varphi}} = \partial/\partial\boldsymbol{\varphi}$.

We discuss an extension of the Semiparametric Locally Efficient Estimation ("SLEE") method of Tchetgen Tchetgen (2014). Specifically, the joint density of the observed data in the multi-group case can be written as

$$f(Y|\boldsymbol{X}, D) f(\boldsymbol{X}|D) \prod_{j=0}^{2} \widetilde{\pi}_j^{1(D=j)} \propto f(Y|\boldsymbol{X}, D) f^*(D|\boldsymbol{X}) f^*(\boldsymbol{X}) \tag{3.15}$$

where $f^*(\boldsymbol{X}) \propto f(\boldsymbol{X})f(D=0|\boldsymbol{X})/f^*(D=0|\boldsymbol{X})$ and

$$\text{logit}(f^*(D=j|\boldsymbol{X})) = \text{logit}(\Pi_j(\boldsymbol{X})) = \text{logit}(P_j(\boldsymbol{X})) - \log\left\{\frac{\widetilde{p}_j(1-\widetilde{\pi}_j)}{\widetilde{\pi}_j(1-\widetilde{p}_j)}\right\}$$

for $j = 1, 2$. We can derive the efficient score of $(\boldsymbol{\theta}, \boldsymbol{\varphi})$ as

$$R(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \left(R_{\boldsymbol{\theta}}(\boldsymbol{\theta}, \boldsymbol{\varphi})^T, \quad R_{\boldsymbol{\varphi}}(\boldsymbol{\theta}, \boldsymbol{\varphi})^T\right), \tag{3.16}$$

where $R_{\boldsymbol{\theta}} = \partial_{\boldsymbol{\theta}}\widetilde{\mu}(\boldsymbol{X}, D; \boldsymbol{\theta}, \boldsymbol{\varphi})\{\text{var}(\epsilon(\boldsymbol{\theta}, \varphi|\boldsymbol{X}, D))\}^{-1}\epsilon(\boldsymbol{\theta}, \boldsymbol{\varphi})$ and

$$R_{\boldsymbol{\varphi}} = \partial_{\boldsymbol{\varphi}}L(\boldsymbol{\varphi}) + \partial_{\boldsymbol{\varphi}}\widetilde{\mu}(\boldsymbol{X}, D; \boldsymbol{\theta}, \boldsymbol{\varphi})\{\text{var}(\epsilon(\boldsymbol{\theta}, \boldsymbol{\varphi}|\boldsymbol{X}, D))\}^{-1}\epsilon(\boldsymbol{\theta}, \boldsymbol{\varphi}).$$

The SLEE method by solving (3.16) is theoretically more efficient than MRLReg, but it is computationally much more difficult. However, simulations in the next section demonstrates that "MRLReg" is competitive in comparison of estimation efficiency compared with "SLEE".

3.2.4 Extension to Binary Secondary Outcome

Our framework can be easily extended to the case when Y is binary. Assume that $\widetilde{\mu}(\boldsymbol{X}, D) = E(Y|\boldsymbol{X}, D) = P(Y=1|\boldsymbol{X}, D)$ and $\mu(\boldsymbol{X}) = P(Y=1|\boldsymbol{X})$ on the logit scale. Let $\text{Odds}(\boldsymbol{X}, D) = P(Y=1|\boldsymbol{X}, D)/P(Y=0|\boldsymbol{X}, D)$ and $\text{Odds}(\boldsymbol{X}) = P(Y=1|\boldsymbol{X})/P(Y=0|\boldsymbol{X})$. Following the derivation of (3.1) in Tchetgen Tchetgen (2014), we can get

$$\text{Odds}(\boldsymbol{X}, D) = \exp\left[\log\{\text{Odds}(\boldsymbol{X})\} + \nu(\boldsymbol{X}, D) - \overline{\nu}(\boldsymbol{X})\right], \tag{3.17}$$

where $\nu(\boldsymbol{X}, D) = \log(\text{Odds}(\boldsymbol{X}, D)/\text{Odds}(\boldsymbol{X}, D=0))$ and

$$\overline{\nu}(\boldsymbol{X}) = \sum_{j=1}^{2}\exp\{\nu(\boldsymbol{X}, D=j)\}P(D=j|\boldsymbol{X}, Y=0) + P(D=0|\boldsymbol{X}, Y=0).$$

If (3.3) holds, we have

$$\log\left\{\frac{\Pi_j^*(\boldsymbol{X})}{\Pi_0^*(\boldsymbol{X})}\right\} = \log\left\{\frac{P_j^*(\boldsymbol{X})}{P_j^*(\boldsymbol{X})}\right\} = m(\boldsymbol{X};\boldsymbol{\varphi}_j), \tag{3.18}$$

where $\Pi_j^*(\boldsymbol{X})$ and $P_j^*(\boldsymbol{X})$ here correspond to $P(D=j|\boldsymbol{X}, Y=0, S=1)$ and $P(D=j|\boldsymbol{X}, Y=0)$, respectively. By setting $\log\{\text{Odds}(\boldsymbol{X})\} = \mu(\boldsymbol{X};\boldsymbol{\beta})$ and $\nu(\boldsymbol{X}, D=j) = \sum_j 1(D=j)g_j(\boldsymbol{X};\boldsymbol{\gamma}_j)$, we have

$$\text{logit}\left\{P(Y=1|D,\boldsymbol{X};\boldsymbol{\theta},\boldsymbol{\varphi})\right\} = \mu(\boldsymbol{X};\boldsymbol{\beta}) + \sum_j 1(D=j)g_j(\boldsymbol{X};\boldsymbol{\gamma}_j) - \overline{\nu}(\boldsymbol{X};\boldsymbol{\gamma}_1,\boldsymbol{\gamma}_2,\boldsymbol{\varphi}) \tag{3.19}$$

with $\boldsymbol{\theta}^T = (\boldsymbol{\beta}^T, \boldsymbol{\gamma}_1^T, \boldsymbol{\gamma}_2^T)$. Similar to $L(\boldsymbol{\varphi})$, we solve the log-likelihood function given by

$$\sum_{i=1}^{N}(1-Y_i)\left[\sum_{j=1}^{2}\{1(D_i=j)\boldsymbol{X}_i^T\boldsymbol{\varphi}_j\} - \log\{1+\sum_{j=1}^{2}\exp(\boldsymbol{X}_i^T\boldsymbol{\varphi}_j)\}\right]. \tag{3.20}$$

Finally, estimating $\boldsymbol{\theta}$ can be done by solving estimation equations based on (3.19).

3.2.5 Extension to Multi-phase Scenario

In this subsection, we extend our regression framework to large-scale multi-group studies with multiple phases. In practice, some studies (e.g., ADNI) collect data across multiple phases, while different phases may follow different sampling schemes. We only consider the case that each subject participates in a single phase, which agrees with the study design of ADNI. For notational simplicity, we consider a three-group study with two phases.

It is assumed that all subjects from different phases follow the same population-level models in terms of $\mu(\boldsymbol{X}) = E(Y|\boldsymbol{X})$, $\widetilde{\mu}(\boldsymbol{X}, D) = E(Y|\boldsymbol{X}, D)$, and $P(D=j|\boldsymbol{X})$, and (3.2) holds for both phases. Similar to (3.5), we have

$$\widetilde{\mu}(\boldsymbol{X}, j) = \mu(\boldsymbol{X}) + \sum_{k\neq j} P(D=k|\boldsymbol{X})\{\widetilde{\mu}(\boldsymbol{X}, j) - \widetilde{\mu}(\boldsymbol{X}, k)\} \tag{3.21}$$

for both phases and each $j = 0, 1, 2$. We still use $\gamma_1(\boldsymbol{X}) = \boldsymbol{X}^T\boldsymbol{\Gamma}_1$, $\gamma_2(\boldsymbol{X}) = \boldsymbol{X}^T\boldsymbol{\Gamma}_2$, and

$\mu(\boldsymbol{X}) = \boldsymbol{X}^T\boldsymbol{\beta}$ to characterize the group difference and target the model at the population level. However, it is assumed that different sampling schemes are used for phases 1 and 2. Let A be the phase from now on, and denote $\Pi_j^{(m)}(\boldsymbol{X}) = P(D = j|\boldsymbol{X}, A = m, S = 1)$ for phase $m = 1, 2$ and group $j = 0, 1, 2$. Thus, (3.3) is given by

$$\frac{\Pi_j^{(m)}(\boldsymbol{X})}{\Pi_0^{(m)}(\boldsymbol{X})} \cdot \frac{\widetilde{\pi}_0}{\widetilde{\pi}_j} = \frac{P_j(\boldsymbol{X})}{P_0(\boldsymbol{X})} \cdot \frac{\widetilde{p}_0^{(m)}}{\widetilde{p}_j^{(m)}} \quad \text{for} \quad m = 1, 2 \quad \text{and} \quad j = 0, 1, 2, \tag{3.22}$$

where $\widetilde{p}_j^{(m)} = P(D = j|S = 1, A = m)$ corresponds to the proportion of group j in the sample at phase m. Subsequently, by assuming a multinomial logistic regression model for $P_j(\boldsymbol{X})$, we have

$$\log\left\{\frac{\Pi_j^{(m)}(\boldsymbol{X})}{\Pi_0^{(m)}(\boldsymbol{X})}\right\} \;=\; \log\left\{\frac{P_j(\boldsymbol{X})}{P_0(\boldsymbol{X})}\right\} + \eta_j^{(m)} = \boldsymbol{X}^T\boldsymbol{\varphi}_j + \eta_j^{(m)}, \tag{3.23}$$

where $\eta_j^{(m)} = \log(\widetilde{p}_0^{(m)}\widetilde{\pi}_j) - \log(\widetilde{p}_j^{(m)}\widetilde{\pi}_0)$ for $m = 1, 2$.

We use a slightly different two-stage estimation procedure to estimate all the parameters of interest. Specifically, in Stage I, we estimate $P_j(\boldsymbol{X})$ for the two phases by combining the observations from both phases. Afterwards, we use the same estimation method in Stage II to estimate additional parameters in $\mu(\boldsymbol{X})$, $\gamma_1(\boldsymbol{X})$, and $\gamma_2(\boldsymbol{X})$. The log pseudo-likelihood function $L(\boldsymbol{\varphi})$ in Stage I is given by

$$\sum_{i=1}^{N}\sum_{m=1}^{2}\left[\sum_{j=1}^{2}\{1(D_i = j)(\boldsymbol{X}_i^T\boldsymbol{\varphi}_j + \eta_j^{(m)})\} - \log\{1 + \sum_{j=1}^{2}\exp(\boldsymbol{X}_i^T\boldsymbol{\varphi}_j + \eta_j^{(m)})\}\right] 1(A_i = m).$$

Under some mild conditions, it can be shown that $\sqrt{n}(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*, \widehat{\boldsymbol{\varphi}} - \boldsymbol{\varphi}_*) \to^L N(\mathbf{0}, \boldsymbol{\Sigma}_*)$, where the covariance matrix $\boldsymbol{\Sigma}_*$ can be approximated by $\widehat{\boldsymbol{\Sigma}}_*$, which is given in the supplements.

3.3 Simulation Studies

We carry out Monte Carlo simulations to evaluate the finite sample performance of five methods including (I) LReg: linear regression without bias correction; (II) LRegD:

linear regression method adjusted for the group status $\boldsymbol{X}_s = (1(D=1), 1(D=2))^T$; (III) IPW: inverse probability weighting approach (Richardson et al., 2007); (IV) SPREG: the retrospective likelihood method in (Lin and Zeng, 2009); (V) MGLReg; and (VI) SLEE: the semiparametric locally efficient estimation method.

3.3.1 Two-SNP Setup

We consider two parts of the simulation. The first part assumes that group difference exists in the genetic effects on the secondary trait. The second part assumes an incorrect specification of the conditional model and a misspecification of the $\gamma_1(\boldsymbol{X}), \gamma_2(\boldsymbol{X})$ (Lin and Zeng, 2009; Zhu et al., 2017; Song et al., 2016). In this setup, one SNP has significant effect on the secondary trait, whereas the other is unrelated.

Setting One The details of the first part are described as follows.

(i) Generate a non-genetic covariate $C \sim N(0,1)$ for each subject.

(ii) Generate two SNP-level genetic variables G_1, G_2 with minor allele frequency (MAF) = 0.3 following a multinomial distribution with frequencies $(p_A^2, 2p_A(1-p_A), (1-p_A)^2)$ for (AA, Aa, aa) respectively, with the Hardy-Weinberg equilibrium assumption under the additive mode of inheritance.

(iii) Generate the primary trait D according to the following multinomial logistic model:

$$\log\left\{\frac{P(D=j|\boldsymbol{X})}{P(D=0|\boldsymbol{X})}\right\} = \boldsymbol{X}^T\boldsymbol{\varphi}_j \text{ for } j=1,2,$$

where $\boldsymbol{X}^T = (1, C, G_1, G_2)$. Subsequently, we can calculate the two dummy variables $1(D=1)$ and $1(D=2)$. Moreover, we choose $\boldsymbol{\varphi}_1 = \boldsymbol{\varphi}_2$ so that the global prevalence of groups 0, 1, and 2 are respectively 10%, 15% and 75%. We also consider a rare disease case with the global prevalence of groups 0, 1 and 2 being 5%, 5% and 90%, respectively.

(iv) Generate the secondary phenotype Y for each subject according to (3.6) as follows:

$$Y = \beta_0 + \boldsymbol{\beta}_1^T \boldsymbol{X} + \sum_{j=1}^{2} \{1(D = j) - P_j(\boldsymbol{X})\}\gamma_j(\boldsymbol{X}) + \epsilon, \tag{3.24}$$

where $\epsilon \sim N(0, \delta)$, $\boldsymbol{\beta}_1^T = (1, 2, 0)$. β_0 and δ are equal to the sample mean and variance of left hippocampi volume from ADNI, respectively. We also set $\gamma_j(\boldsymbol{X}_i) = \boldsymbol{X}_i^T \boldsymbol{\Gamma}_j$ for $j = 1, 2$ with $\boldsymbol{\Gamma}_1 = (-2, -1, -1, -1)^T$ and $\boldsymbol{\Gamma}_2 = (1, 1, 1, 1)^T$.

(v) Repeat steps (i)-(iv) to generate $(Y, \boldsymbol{X}, D)$ until we obtain a total of $N = 500,000$ observations as the whole population. Then, we randomly select 500, 1000, and 500 subjects from the $D = 0$, $D = 1$, and $D = 2$ groups to build a non-random three-group sample.

Setting Two

(i) Generate $\boldsymbol{X}^T = (1, C, G_1, G_2)$ as setting one.

(ii) Generate the secondary phenotype Y for each subject according to

$$Y = \beta_0 + \boldsymbol{\beta}_1^T \boldsymbol{X} + \epsilon, \tag{3.25}$$

and we still have $\epsilon \sim N(0, \delta)$, $\boldsymbol{\beta}_1^T = (1, 2, 0)$, and the same (β_0, δ) as setting one.

(iii) Simulate the primary trait D using a multinomial model given by

$$\log\left\{\frac{P(D = j|\boldsymbol{X}, Y)}{P(D = 0|\boldsymbol{X}, Y)}\right\} = (\boldsymbol{X}^T, Y)\widetilde{\boldsymbol{\varphi}}_j \text{ for } j = 1, 2,$$

and we also vary $\widetilde{\boldsymbol{\varphi}}_1, \widetilde{\boldsymbol{\varphi}}_2$ to get the global group prevalences to be (10%,15%,75%) and (5%,5%,90%) for the rare case, respectively.

(iv) Repeat steps 1-3 until the sample size reaches 500, 000 and then sample 500 (D=0), 1000 (D=1) and 500 (D=2) observations from the above large pool of subjects.

Table 3.1: Estimation biases, variances, and 95% coverage rates of $\widehat{\beta}_G$ for $p_A = 0.3$

		Setting1			Setting2		
		Absolute Bias	Variance	Coverage	Absolute Bias	Variance	Coverage
$\beta_{G_1} = 2$	LReg	0.8526	1.06×10^{-2}	0.012	0.1774	9.22×10^{-3}	0.572
	LRegD	0.5639	2.79×10^{-2}	0.066	0.8633	7.59×10^{-3}	0.000
	IPW	0.0848	1.94×10^{-2}	0.945	0.1180	2.14×10^{-2}	0.945
	SPREG	1.2001	1.78×10^{-1}	0.000	0.0889	1.13×10^{-2}	0.946
	MGLReg ($\widetilde{p}_0 = .1, \widetilde{p}_1 = .15$)	0.0615	2.69×10^{-3}	0.969	0.0987	1.49×10^{-2}	0.946
	SLEE ($\widetilde{p}_0 = .1, \widetilde{p}_1 = .15$)	0.0613	2.63×10^{-3}	0.970	0.0986	1.48×10^{-2}	0.948
	MGLReg ($\widetilde{p}_0 = .05, \widetilde{p}_1 = .15$)	0.0633	3.21×10^{-3}	0.954	0.1014	1.37×10^{-2}	0.936
	SLEE ($\widetilde{p}_0 = .05, \widetilde{p}_1 = .15$)	0.0631	3.20×10^{-3}	0.956	0.1006	1.36×10^{-2}	0.935
	MGLReg ($\widetilde{p}_0 = .15, \widetilde{p}_1 = .15$)	0.0671	2.75×10^{-3}	0.960	0.1053	1.77×10^{-2}	0.926
	SLEE ($\widetilde{p}_0 = .15, \widetilde{p}_1 = .15$)	0.661	2.69×10^{-3}	0.961	0.1048	1.74×10^{-2}	0.928
	MGLReg ($\widetilde{p}_0 = .1, \widetilde{p}_1 = .1$)	0.1008	6.26×10^{-3}	0.884	0.1065	1.58×10^{-2}	0.914
	SLEE ($\widetilde{p}_0 = .1, \widetilde{p}_1 = .1$)	0.0993	6.15×10^{-3}	0.886	0.1029	1.56×10^{-2}	0.918
	MGLReg ($\widetilde{p}_0 = .1, \widetilde{p}_1 = .2$)	0.0955	3.15×10^{-3}	0.854	0.0982	1.10×10^{-2}	0.956
	SLEE ($\widetilde{p}_0 = .1, \widetilde{p}_1 = .2$)	0.0942	3.07×10^{-3}	0.886	0.0972	1.04×10^{-2}	0.960
$\beta_{G_2} = 0$	LReg	0.8483	1.08×10^{-2}	0.000	0.1478	1.11×10^{-2}	0.776
	LRegD	0.9744	2.04×10^{-2}	0.000	0.3744	6.79×10^{-3}	0.014
	IPW	0.0752	1.73×10^{-2}	0.944	0.1137	2.55×10^{-2}	0.950
	SPREG	0.7418	1.04×10^{-1}	0.112	0.0994	1.53×10^{-2}	0.954
	MGLReg ($\widetilde{p}_0 = .1, \widetilde{p}_1 = .15$)	0.0655	6.80×10^{-3}	0.954	0.1050	1.99×10^{-2}	0.952
	SLEE ($\widetilde{p}_0 = .1, \widetilde{p}_1 = .15$)	0.0644	6.55×10^{-3}	0.954	0.1036	1.92×10^{-2}	0.952
	MGLReg ($\widetilde{p}_0 = .05, \widetilde{p}_1 = .15$)	0.0868	9.86×10^{-3}	0.868	0.1050	1.99×10^{-2}	0.952
	SLEE ($\widetilde{p}_0 = .05, \widetilde{p}_1 = .15$)	0.0851	9.75×10^{-3}	0.870	0.1036	1.92×10^{-2}	0.952
	MGLReg ($\widetilde{p}_0 = .15, \widetilde{p}_1 = .15$)	0.0714	5.62×10^{-3}	0.924	0.1070	1.76×10^{-2}	0.948
	SLEE ($\widetilde{p}_0 = .15, \widetilde{p}_1 = .15$)	0.0706	5.53×10^{-3}	0.928	0.1049	1.99×10^{-2}	0.950
	MGLReg ($\widetilde{p}_0 = .1, \widetilde{p}_1 = .1$)	0.0945	7.53×10^{-3}	0.846	0.0987	2.61×10^{-2}	0.930
	SLEE ($\widetilde{p}_0 = .1, \widetilde{p}_1 = .1$)	0.0947	7.38×10^{-3}	0.848	0.1043	2.54×10^{-2}	0.932
	MGLReg ($\widetilde{p}_0 = .1, \widetilde{p}_1 = .2$)	0.0938	6.39×10^{-3}	0.844	0.1023	1.91×10^{-2}	0.946
	SLEE ($\widetilde{p}_0 = .1, \widetilde{p}_1 = .2$)	0.0897	5.95×10^{-3}	0.850	0.1019	1.86×10^{-2}	0.946

Tables 3.1 and 3.2 present the simulation results under the first and second simulation setups. They include the mean absolute biases and the variances of $\widehat{\beta}_G$ and, their 95% confidence interval coverage rates based on the 1,000 Monte Carlo samples for all six methods. Both LReg and LRegD perform poorly in correcting the sampling bias for both settings. Under the first setting, MGLReg and SLEE introduced in this chapter have the smallest estimation bias. The SLEE performs slightly better than MGLReg, but the difference is not substantial. The IPW achieves a comparable performance with MGLReg, whereas our method is more efficient under both settings. The likelihood-based approach SPREG does not work in the first part, since it highly depends on the correct specification of the conditional model. For the second part, MGLReg and SLEE provide competitive estimation results with SPREG, especially in the rare disease case. On the other hand, as we misspecify $(\widetilde{p}_0, \widetilde{p}_1)$, both MGLReg and SLEE perform acceptably stable under different global prevalence settings.

Table 3.2: Estimation biases, variances, and 95% coverage rates of $\widehat{\beta}_G$ for rare disease case

		Setting1			Setting2		
		Absolute Bias	Variance	Coverage	Absolute Bias	Variance	Coverage
$\beta_{G_1} = 2$	LReg	1.5466	7.14×10^{-3}	0.000	0.6095	7.49×10^{-3}	0.102
	LRegD	0.7638	2.37×10^{-2}	0.004	1.0535	6.54×10^{-3}	0.000
	IPW	0.0686	5.68×10^{-3}	0.832	0.1859	3.25×10^{-2}	0.640
	SPREG	0.1546	1.22×10^{-1}	0.896	0.1486	9.82×10^{-2}	0.891
	MGLReg ($\widetilde{p}_0 = .05, \widetilde{p}_1 = .05$)	0.0552	4.86×10^{-3}	0.916	0.1139	2.14×10^{-2}	0.928
	SLEE ($\widetilde{p}_0 = .05, \widetilde{p}_1 = .05$)	0.0552	4.85×10^{-3}	0.920	0.1081	1.92×10^{-2}	0.930
	MGLReg ($\widetilde{p}_0 = .05, \widetilde{p}_1 = .1$)	0.0726	4.42×10^{-3}	0.868	0.1313	2.84×10^{-2}	0.911
	SLEE ($\widetilde{p}_0 = .05, \widetilde{p}_1 = .1$)	0.0720	4.39×10^{-3}	0.872	0.1308	2.59×10^{-2}	0.912
	MGLReg ($\widetilde{p}_0 = .1, \widetilde{p}_1 = .05$)	0.0709	3.83×10^{-3}	0.880	0.1293	2.47×10^{-2}	0.912
	SLEE ($\widetilde{p}_0 = .1, \widetilde{p}_1 = .05$)	0.0714	3.81×10^{-3}	0.884	0.1252	2.38×10^{-2}	0.916
$\beta_{G_2} = 0$	LReg	1.0773	1.34×10^{-2}	0.000	0.3857	8.91×10^{-3}	0.390
	LRegD	1.0959	8.33×10^{-3}	0.000	0.5367	7.35×10^{-3}	0.004
	IPW	0.0751	7.88×10^{-3}	0.850	0.1536	3.42×10^{-2}	0.950
	SPREG	0.1376	1.38×10^{-1}	0.884	0.1349	5.10×10^{-2}	0.921
	MGLReg ($\widetilde{p}_0 = .1, \widetilde{p}_1 = .15$)	0.0712	6.80×10^{-3}	0.970	0.1270	2.30×10^{-2}	0.946
	SLEE ($\widetilde{p}_0 = .1, \widetilde{p}_1 = .15$)	0.0710	6.55×10^{-3}	0.972	0.1240	2.18×10^{-2}	0.950
	MGLReg ($\widetilde{p}_0 = .05, \widetilde{p}_1 = .1$)	0.0806	9.94×10^{-3}	0.926	0.1448	3.37×10^{-2}	0.938
	SLEE ($\widetilde{p}_0 = .05, \widetilde{p}_1 = .1$)	0.0797	9.70×10^{-3}	0.932	0.1399	3.18×10^{-2}	0.940
	MGLReg ($\widetilde{p}_0 = .1, \widetilde{p}_1 = .05$)	0.0795	8.87×10^{-3}	0.930	0.1432	3.01×10^{-2}	0.942
	SLEE ($\widetilde{p}_0 = .1, \widetilde{p}_1 = .05$)	0.0793	8.85×10^{-3}	0.932	0.1429	2.96×10^{-2}	0.945

More details are given in Tables 3.1 and 3.2. In terms of the computation efficiency, MGLReg is about 10-times faster than SLEE. Therefore, we choose MGLReg to do the large-scale ADNI data analysis.

3.3.2 Multiple-SNP Setup

To better mimic the real-world GWAS analysis, we use the same simulation settings as those for the two-SNP setup except adopting a multiple-SNP setup with in total 500 SNPs and randomly sampling 10 SNPs as causal SNPs with effect size being 0:5. For details, please refer to the supplementary document.

Table 3.3 presents the mean absolute biases, the mean estimation variances and their 95% confidence interval coverage rates based on 100 Monte Carlo samples of both the causal and non-causal SNPs for all methods. Table 3.3 shows that our method MGLReg can detect more causal SNPs (higher mean coverage rates) compared to the other methods in both settings, demonstrating that our method is more robust against biased sampling and less sensitive to model misspecification. Compared to the two-SNP setup, IPW is more biased especially for setting two, whereas our method is much more stable. SPREG does not perform well in

this case even for setting two, which confirms our conclusion that SPREG highly depends on the correct specification of the conditional model. For SNPs not associated with secondary phenotype, MGLReg performs similar to others. It means that it does not overestimate the genetic effects of non-causal SNPs even with higher model complexity.

Table 3.3: Mean estimation biases, variances, and 95% coverage rates of Causal and Non-causal SNPs

		Setting1			Setting2		
		Absolute Bias	Variance	Coverage	Absolute Bias	Variance	Coverage
Causal SNPs	LReg	0.2996	1.16×10^{-2}	0.128	0.2032	9.02×10^{-3}	0.512
	LRegD	0.3042	1.02×10^{-1}	0.220	0.3691	6.38×10^{-3}	0.000
	IPW	0.0693	7.61×10^{-3}	0.902	0.1772	5.46×10^{-2}	0.648
	SPREG	0.2998	1.17×10^{-1}	0.132	0.1534	3.57×10^{-2}	0.904
	MGLReg	0.0557	4.83×10^{-3}	0.944	0.1075	1.14×10^{-2}	0.956
Non-Causal SNPs	LReg	0.0674	7.07×10^{-3}	0.923	0.1464	1.01×10^{-2}	0.929
	LRegD	0.0576	4.01×10^{-3}	0.943	0.0961	6.91×10^{-2}	0.933
	IPW	0.0700	7.59×10^{-3}	0.907	0.1898	5.68×10^{-2}	0.645
	SPREG	0.0693	8.39×10^{-3}	0.940	0.1302	3.39×10^{-2}	0.937
	MGLReg	0.0549	5.06×10^{-3}	0.951	0.0896	1.67×10^{-2}	0.947

3.4 The Alzheimer's Disease Neuroimaging Initiative Data

We apply the MGLReg method to the ADNI data set. The main goal of this data analysis is to search for genetic patterns that are associated with local hippocampal changes, while correcting for the selection bias associated with ascertainment in multi-group studies.

3.4.1 GWAS analysis

The 299 subjects with normal cognition (NC), 553 with MCI and 185 with AD build the final sample data, where 712 of them are from ADNI 1 with the other 325 from ADNI 2 and GO. The secondary outcome Y used in the experiment are the logarithm of the left and right hippocampi volumes divided by the whole brain volume. The 6,017,259 SNPs after quality control are analyzed, and the genetic factor at each individual SNP is coded as $0, 1$ and 2. To correct for the population stratification, the top three principal components (PCs) of the whole-genome data are included as covariates (Price et al., 2006). We also add a dummy variable for distinguishing ADNI1 from (ADNI2, ADNIGO), since different imaging protocols were used in ADNI1 and (ADNI2, ADNIGO), which may affect the volume

segmentation results. We apply two-sample T-test to test the difference between ADNI1 and (ADNI2, ADNIGO), whose $p-$value is smaller than $2e-16$. Thus, a significant difference exists between the distribution of Y for ADNI1 and that for (ADNI2, ADNIGO) according to the boxplot in the supplements. The details of data description and processing procedures are discussed in supplementary material.

In this data analysis, $D = 0$, 1, and 2 represent AD, MCI and NC, respectively. The global prevalence of AD within people older than 65 is more than 10% (Thies and Bleiler, 2012) while MCI is between 10% and 20% (Kim et al., 2015). We compare four different combinations of $(\widetilde{p}_0, \widetilde{p}_1)$, $(0.1, 0.15), (0.1, 0.2), (0.15, 0.15)$, and $(0.15, 0.2)$ for our proposed method, since the prevalences of AD and MCI vary with patients getting old, and the chance of developing MCI and AD increases as adults age.

3.4.2 Results

Table 3.4 presents the most significant pairs of SNPs combined with the regions of interest detected by LReg, where significant SNPs are selected according to the 5×10^{-8} $p-$value threshold for both the left and right hippocampi. The p-values of these SNPs by MGLReg with different $(\widetilde{p}_0, \widetilde{p}_1)$ selections are also provided. Those p-values smaller than 5×10^{-8} are marked.

The SNP rs429358, related to gene APOE, is detected as the most significant SNP for both left and right hippocampi by both LReg and MGLReg. Specifically, rs429358 has significant genetic effects on the volume size of left hippocampi since its $p-$value is consistently smaller than the $5e^{-8}$ threshold with different combinations of $(\widetilde{p}_0, \widetilde{p}_1)$. This result agrees with the previous findings (Shen et al., 2010; Kim et al., 2002; Lu et al., 2011; Kim et al., 2015). Another significant SNP rs769449, also in APOE region, has competitive significancy with rs429358 for both left and right hippocampi, which was found to be associated with cerebrospinal fluid (CSF) tau (Cruchaga et al., 2013) and verbal memory (Arpawong et al., 2017). Therefore, our results may prove that rs769449 may have potential effects on the hippocampi volumes. Other significant SNPs detected by LReg are not stably significant

Table 3.4: Top SNPs and $p-$values for association tests with the left and right hippocampus volumes

Left hippocampus

SNPs	chr	common effect					interaction				
		LReg	MGLReg				LReg	MGLReg			
			$(0.2, 0.15)$	$(0.15, 0.15)$	$(0.2, 0.1)$	$(0.15, 0.1)$		$(0.2, 0.15)$	$(0.15, 0.15)$	$(0.2, 0.1)$	$(0.15, 0.1)$
rs429358	19	1.76e-11	3.79e-11	2.00e-10	5.01e-09	3.48e-08	0.797	0.938	0.883	0.766	0.732
rs769449	19	5.21e-10	1.38e-09	5.15e-09	6.09e-08	2.96e-07	0.874	0.718	0.642	0.615	0.554
rs10414043	19	6.34e-10	4.44e-09	1.72e-08	1.68e-07	8.18e-07	0.827	0.700	0.633	0.595	0.542
rs73052335	19	1.39e-09	1.55e-08	5.70e-08	5.45e-07	2.47e-06	0.751	0.643	0.582	0.529	0.484
rs59007384	19	3.77e-08	1.38e-05	4.96e-05	6.56e-04	1.86e-03	0.406	0.771	0.742	0.661	0.655

Right hippocampus

SNPs	chr	common effect					interaction				
		LReg	MGLReg				LReg	MGLReg			
			$(0.2, 0.15)$	$(0.15, 0.15)$	$(0.2, 0.1)$	$(0.15, 0.1)$		$(0.2, 0.15)$	$(0.15, 0.15)$	$(0.2, 0.1)$	$(0.15, 0.1)$
rs429358	19	1.17e-10	3.82e-09	1.77e-08	4.69e-08	3.04e-06	0.089	0.325	0.324	0.223	0.239
rs769449	19	2.37e-09	4.99e-10	1.38e-09	3.24e-08	1.20e-07	0.109	0.286	0.287	0.205	0.221
rs10414043	19	2.35e-09	9.55e-10	2.70e-09	5.70e-08	2.09e-07	0.105	0.297	0.302	0.204	0.221
rs73052335	19	3.76e-09	3.82e-09	1.08e-08	2.01e-07	7.21e-07	0.100	0.260	0.263	0.171	0.185
rs6857	19	5.20e-09	4.31e-07	1.86e-06	3.18e-05	1.33e-04	0.253	0.730	0.701	0.511	0.516
rs283812	19	2.92e-08	2.24e-06	7.29e-06	1.23e-04	3.89e-04	0.116	0.121	0.139	0.124	0.150
rs59007384	19	7.81e-09	1.89e-05	6.51e-05	7.98e-04	2.42e-03	0.106	0.747	0.768	0.653	0.708

when the population rates vary according to the results of our approach. For example, rs59007384 (associated with gene TOMM40) is related to the progression from MCI status to AD (Cervantes et al., 2011). The higher group proportion of AD in the sample data may result in the significant p-value by LReg. However, our method MGLReg indicates that rs59007384 may not be significantly related with the hippocampi volume sizes in the whole population, especially the group of normal people.

Figure 3.1 presents the heatmaps of $\log_{10}(p)-$value for SNPs rs429358, rs769449, and rs59007384 using MGLReg, with $\widetilde{p}_0$ and $\widetilde{p}_1$ varying within $[0.1, 0.35]$ and $[0.1, 0.65]$ respectively, demonstrating a dynamic change of significance over various MCI and AD prevalence rates in the whole population. We introduce the Significance Prevalence Heatmap (SPH) by using ellipse contours corresponding to different $p-$value thresholds to determine the population prevalence range for the significance of a specific SNP. For instance, if $\widetilde{p}_0 + \widetilde{p}_1$ is smaller than 0.5, then within the given $(\widetilde{p}_0, \widetilde{p}_1)$ range, rs429358 is significant for the left hippocampi as $\widetilde{p}_0 + 2.625 * \widetilde{p}_1 > 0.4045$ and for the right hippocampi as $\widetilde{p}_0 + 3.138 * \widetilde{p}_1 > 0.596$; rs769449 is

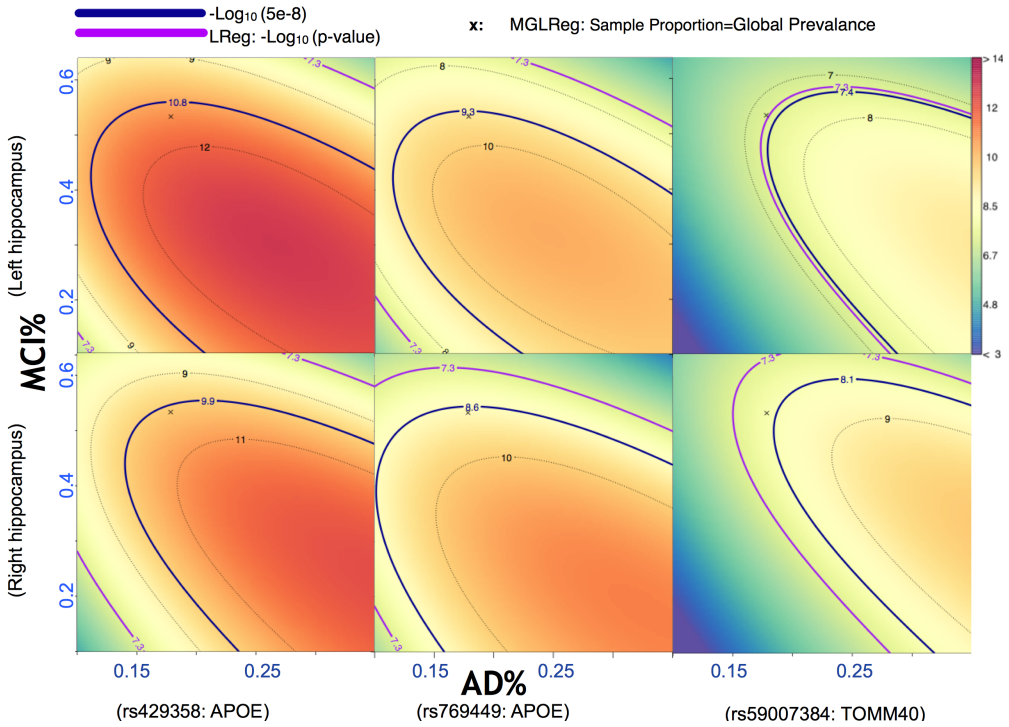

Figure 3.1: The heatmaps of $-\log_{10}(p)$-value for three selected SNPs by MGLReg with different global AD and MCI prevalence rates in the whole population

significant for the left hippocampi as $\widetilde{p}_0 + 2.70 * \widetilde{p}_1 > 0.478$ and for the right hippocampi as $\widetilde{p}_0 + 3.5 * \widetilde{p}_1 > 0.534$.

To more clearly show how the global prevalence rate $(\widetilde{p}_0, \widetilde{p}_1)$ influences the genetic effects, we plot the density curves of the $-\log_{10}(p)$-values of 50 SNPs in the APOE region by LReg and MGLReg with different $(\widetilde{p}_0, \widetilde{p}_1)$ combinations (Figure 3.2). The curves shift to left as $(\widetilde{p}_0, \widetilde{p}_1)$ decreases. It indicates that most significant SNPs in this region detected by LReg are considered unimportant in normal people. Only those SNPs jointly detected by both LReg and MGLReg with all $(\widetilde{p}_0, \widetilde{p}_1)$ settings have significant population-level genetic effects on the hippocampi volume size.

Since the genetic measurements were on different platforms, we do an interaction analysis to test its potential differences and consequences on inference. Specifically, we repeat the experiment above, but adding an interaction term between phase status and genetic factor into the covariates set. We include the p-values of testing the interaction term for the top

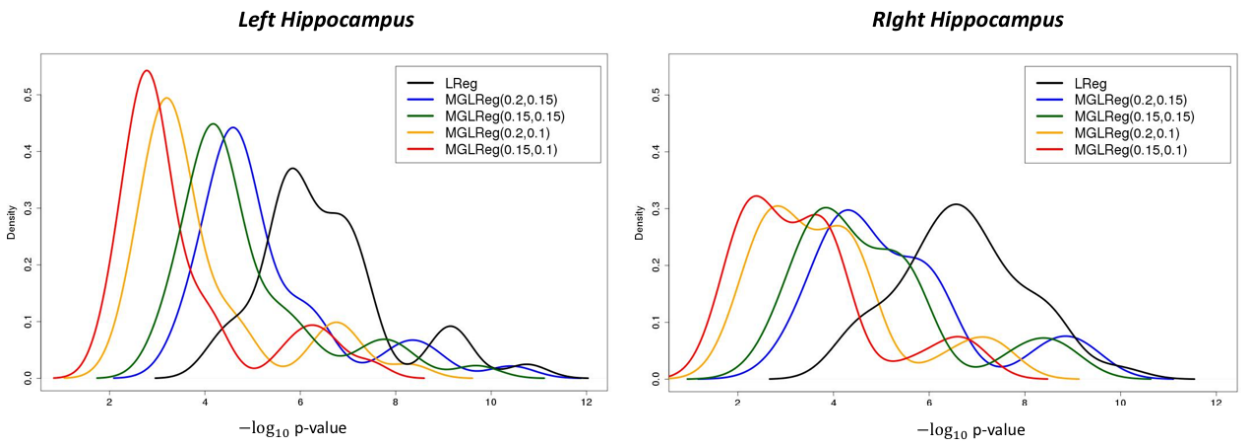

Figure 3.2: The density curves of $-\log_{10}(p)$-values of top 50 APOE-region SNPs by each method for the left and right hippocampus volumes

SNPs in Table 3.4. We observe that the genetic data acquired at the two phases do not have significant difference based on the $p-$values. Figures 3.3 and 3.4 present the Manhattan plots of the GWAS results based on the left and right hippocampi by all the 6,017,259 SNPs to give a global view of the genetic effects and their variation as the global prevalence rate varies.

3.5 Discussion

The aim of this chapter is to develop a general regression framework based on the conditional model for the secondary outcome given the multi-group status and covariates and its relationship with the population regression of interest of the secondary outcome given covariates. It allows us to reduce the effect of sampling bias on the association between a certain genetic factor G and secondary trait Y in multi-group studies. Our method shares a similar idea with the traditional weighted likelihoods method such as IPW in correcting the weights of subjects in multiple groups, but it outperforms IPW in terms of smaller estimation bias and type-I error rate. The GWAS experiment clearly demonstrates how the global prevalence rates influence the effects of covariates on the secondary outcome. Our MGLReg reduces to standard linear regression when the sample proportions are the same as the global ones. Our experiment provides more evidence that rs429358 and rs769449 have whole-population level genetic effects on the volume sizes of left and right hippocampi. On the other hand, other top SNPs detected by LReg may be caused by the sampling bias by

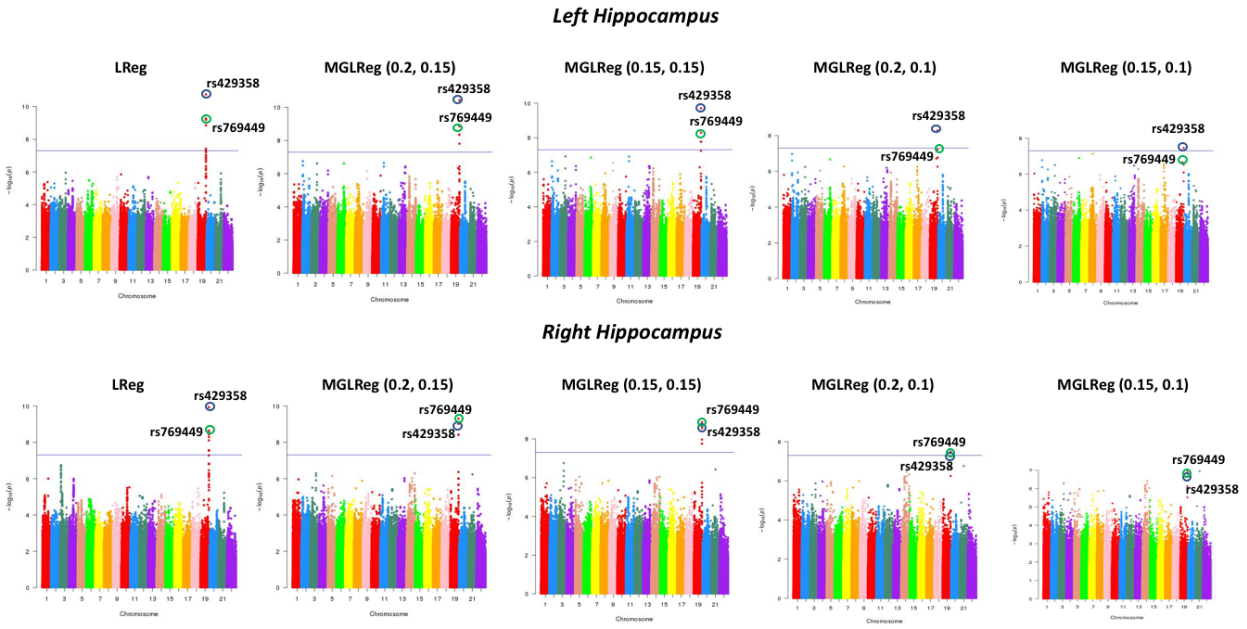

Figure 3.3: The Manhattan plots of the $-\log(p)-$values by LReg and MGLReg on all 22 chromosomes for the left and right hippocampus volumes

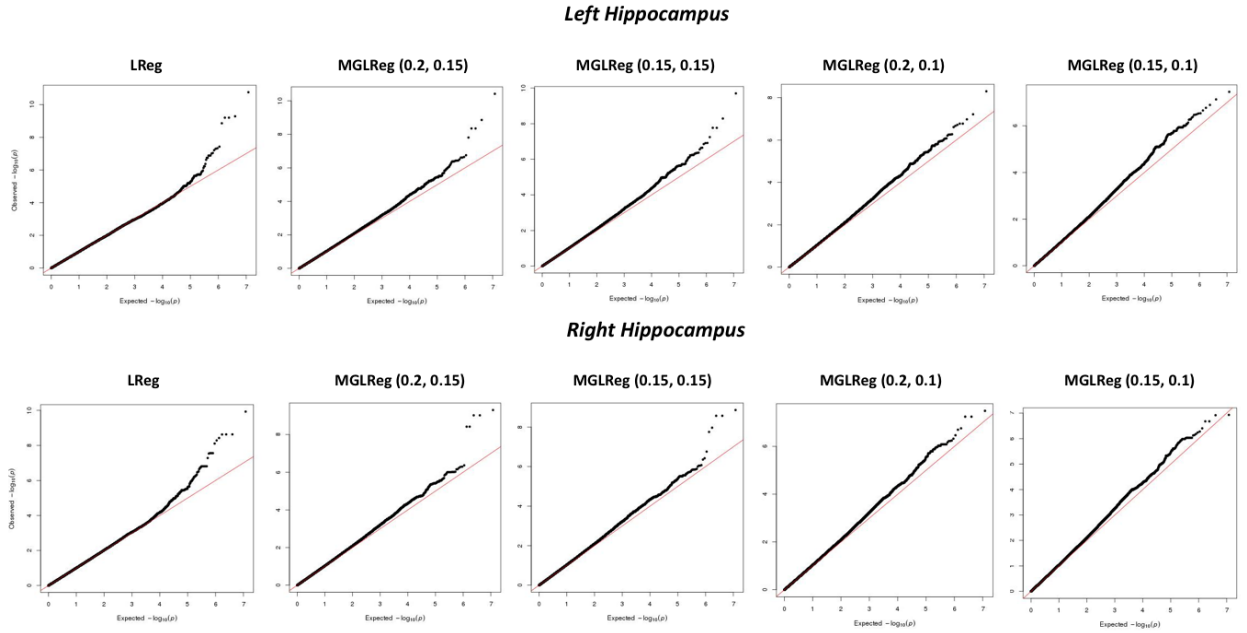

Figure 3.4: The Manhattan plots of the $-\log(p)-$values by LReg and MGLReg on all 22 chromosomes for the left and right hippocampus volumes

our method.

CHAPTER 4: GRAPH-BASED SEMI-SUPERVISED LEARNING WITH NONIGNORABLE NONRESPONSES

4.1 Introduction

Graph-based semi-supervised learning problem has been increasingly studied due to more and more real graph datasets. The problem is to predict all the unlabelled nodes in the graph based on only a small subset of nodes being observed. A popular method is to use the graph Laplacian regularization to learn node representations, such as label propagation (Zhu et al., 2003) and manifold regularization (Belkin et al., 2006). Recently, attention has shifted to the learning of network embeddings (Mikolov et al., 2013; Perozzi et al., 2014; Tang et al., 2015; Grover and Leskovec, 2016; Yang et al., 2016; Kipf and Welling, 2016; Defferrard et al., 2016). Almost all existing methods assume that the labelled nodes are randomly selected. However, the probability of missingness may depend on the unobserved data after conditioning on the observed data. That is, non-responses may be missing not at random (MNAR). Ignoring nonignorable nonresponses may be unable to capture the representativeness of remaining samples, leading to significant estimation bias.

Modeling non-ignorable missingness is challenging because the MNAR mechanism is usually unknown and may require additional model identifiability assumptions (Chen, 2001; Qin et al., 2002; Tang et al., 2014). A popular method assigns the inverse of estimated response probabilities as weights to the observed nodes (Robins et al., 1995; Carpenter et al., 2006), but these procedures are designed for the missing at random (MAR) mechanism instead of MNAR. Another method is to impute missing data by using observed data (Rubin, 1976; Schafer and Schenker, 2000; Little and Rubin, 2019). Some more advanced methods (Zhao et al., 2013; Tang et al., 2014) have been proposed to estimate the non-ignorable missingness using external data (Kim and Yu, 2011), but such data is often unavailable in

many applications, making these methods infeasible. Moreover, all these methods are built on simple regressions and are not directly applied to graphs.

In this chapter, we develop a Graph-based joint model with Nonignorable Missingness (GNM) by assigning inverse response probability to labelled nodes when estimating the target classifier or regression. To model the non-ignorable missingness, we propose a deep learning based exponential tilting model to utilize the strengths of neural networks in function approximation and representation learning. The main contributions can be summarized as follows:

- To the best of our knowledge, we are the first to consider the graph-based semi-supervised learning problem in the presence of non-ignorable nonresponse and try to solve the problem from the perspective of missing data.

- We propose a novel joint estimation approach by integrating the inverse weighting framework with a modified loss function based on the imputation of non-response, which is easy to implement in practice and robust to the normality assumption when the node response is continuous.

- We use gradient descent (GD) algorithm to learn all the parameters, which works for traditional regression model as well as for modern deep graphical neural networks.

- We examine the finite sample performance of our methods by using both simulation and real data experiments, demonstrating the necessity of 'de-biasing' in acquiring unbiased prediction results on the testing data under the non-ignorable nonresponse setting.

4.2 Model Description

Let $G = (V, E, A)$ be a weighted graph, where $V = \{v_1, \ldots, v_N\}$ denotes the vertex set of size $|V| = N$, E contains all the edges, and A is an $N \times N$ adjacency matrix. The N vertexes make up the whole population with only a small subset of vertexes being labelled. We introduce some important notations as follows:

(i). $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T \in R^{N\times p}$ is a fully observed input feature matrix of size $N \times p$ with each $x_i \in R^p$ being a $p \times 1$ feature vector at vertex v_i.

(ii). $Y = (y_1, y_2, \ldots, y_N)^T$ is a vector of vertex responses, which is partially observed subject to missingness, and y_i can be either categorical or continuous.

(iii). $A \in R^{N\times N}$ is the adjacency matrix (binary or weighted), which encodes node similarity and network connectivity. Specifically, a_{ij} represents the edge weight between vertexes v_i and v_j.

(iv). $r_i \in \{0, 1\}$ is a "labeling indicator", that is y_i is observed if and only if $r_i = 1$. Let $R = \{1, \ldots, n\}$ denote the set of labelled vertexes and $R^c = \{n + 1, \ldots, N\}$ defines the subsample of non-respondents for which the vertex label is missing.

(v) $\mathscr{G}^A(\mathbf{x}; \theta_g) \in R^{N\times q}$ denotes a $q \times 1$ vector of unknown function of $\mathbf{x}$, which can be a deep neural network incorporating the network connectivity A.

In this chapter, we consider an non-ignorable response mechanism, where the indicator variable r_i depends on y_i (which is unobserved when $r_i = 0$). It is assumed that r_i follows a Bernoulli distribution as follows:

$$r_i|(y_i, h(x_i; \theta_h)) \sim \text{Bernoulli}(\pi_i), \tag{4.1}$$

where $h(x_i; \theta_h)$ is an unknown parametric function of x_i and $\pi(y_i, h(x_i; \theta_h)) = P(r_i = 1|y_i, h(x_i; \theta_h))$ is the probability of missingness for y_i. Given $\mathscr{G}^A(\mathbf{x}; \theta_g)$, y_i and y_j are assumed to be independent and given y_i and $h(x_i; \theta_h)$, r_i and r_j are assumed to be independent for $i \neq j$. Furthermore, an exponential tilting model is proposed for π_i as follows:

$$\pi(y_i, h(x_i; \theta_h)) = \pi(y_i, h(x_i; \theta_h); \alpha_r, \gamma, \phi) = \frac{\exp\{\alpha_r + \gamma^T h(x_i; \theta_h) + \phi y_i\}}{1 + \exp\{\alpha_r + \gamma^T h(x_i; \theta_h) + \phi y_i\}}. \tag{4.2}$$

Our question of interest is to unbiasedly learn an outcome model $Y|\mathbf{x}$. Without loss of

generality, when y is continuous, we consider a linear model given by

$$Y = \alpha + \mathscr{G}^A(\mathbf{x}; \theta_g)\beta + \epsilon, \tag{4.3}$$

where $\epsilon = (\epsilon_1, \cdots, \epsilon_N)^T \sim N(\mathbf{0}, \sigma^2 I)$ and $\epsilon \perp \mathbf{x}$ is the error term with zero unconditional mean, that is, $E(\epsilon_i) = 0$. In this case, dropping out missing data can lead to strongly biased estimates when r depends on y. The parameter estimates will not be consistent since $E\{\epsilon_i | r_i = 1\}$ and $E\{\epsilon_i \mathscr{G}^A(\mathbf{x}; \theta_g)_i | r_i = 1\}$ are not zero. The missing values could not be imputed even if we would have consistent estimates since

$$\begin{aligned} E\{y_i | r_i = 0, \mathscr{G}^A(\mathbf{x}; \theta_g)_i; \alpha, \beta\} &= \frac{E\{y_i(1 - r_i) | \mathscr{G}^A(\mathbf{x}; \theta_g)_i; \alpha, \beta\}}{1 - P(r_i = 1 | \mathscr{G}^A(\mathbf{x}; \theta_g)_i; \alpha, \beta)} \\ = \quad \alpha + \beta^T \mathscr{G}^A(\mathbf{x}; \theta_g)_i - \frac{\mathrm{cov}(y_i, \pi_i | \mathscr{G}^A(\mathbf{x}; \theta_g)_i; \alpha, \beta)}{1 - E(\pi_i | \mathscr{G}^A(\mathbf{x}; \theta_g)_i; \alpha, \beta)} &\neq \alpha + \beta^T \mathscr{G}^A(\mathbf{x}; \theta_g)_i. \end{aligned} \tag{4.4}$$

When y is a K-class discrete variable, we consider an multicategorical logit model as follow:

$$P(y_i = k | \mathscr{G}^A(\mathbf{x}; \theta_g)_i; \alpha_k, \beta_k) = \exp(\alpha_k + \beta_k^T \mathscr{G}^A(\mathbf{x}; \theta_g)_i) / \sum_{j=1}^{K} \exp(\alpha_j + \beta_j^T \mathscr{G}^A(\mathbf{x}; \theta_g)_i) \quad \forall k \tag{4.5}$$

Therefore, we can define a joint model of (4.2) and (4.3) (or (4.2) and (4.5)), called Graph-based joint model with Nonignorable Missingness (GNM) to obtain the unbiased estimation of $Y|\mathbf{x}$.

4.3 Estimation

We examine several important properties, such as identifiability, of GNM and its estimation algorithm in this section.

4.3.1 Identifiability

We consider the identifiability property of GNM. Let $Y = (y_{obs}^T, y_{mis}^T)^T$ and $J = (R, R^c)$. The joint probability density function (pdf) of the observed data is given by

$$f(y_{obs}, J|\mathbf{x}) = f(y_1, y_2 \ldots, y_n, r_1, \ldots, r_N|\mathbf{x}) = \prod_{i=1}^{n} f(y_i, r_i|\mathbf{x}) \prod_{i=n+1}^{N} \int f(y_i, r_i|\mathbf{x}) dy_i. \quad (4.6)$$

Based on the assumptions of $r_i|(y_i, h(x_i))$ and $y_i|\mathscr{G}^A(\mathbf{x}; \theta_g)_i$, (4.6) is equivalent to

$$\prod_i [P(r_i = 1|y_i, h(x_i; \theta_h)) f(y_i|\mathscr{G}^A(\mathbf{x}; \theta_g)_i)]^{r_i} [1 - \int P(r_i = 1|y, h(x_i; \theta_h)) f(y|\mathscr{G}^A(\mathbf{x}; \theta_g)_i) dy]^{1-r_i}. \quad (4.7)$$

The GNM model is called identifiable if for different sets of parameters (θ_h, θ_g), $P(r_i = 1|y_i, h(x_i; \theta_h)) f(y_i|\mathscr{G}^A(\mathbf{x}; \theta_g)_i)$ are different functions of $(y_i, \mathbf{x})$. The identifiability implies that in a positive probability, the global maximum of (4.7) is unique.

However, identifiability may fail for many neural network models. For example, the identifiability of parameters in (4.2) is one of the necessary conditions for model identifiability, which can fail for the Relu network. Specifically, we have

$$\text{Logit}[P(r_i = 1|y_i, h(z_i; \beta_r)); \gamma] = \alpha_r + \gamma \text{Relu}(z_i \beta_r) + \phi y_i = \text{Logit}[P(r_i = 1|y_i, h(z_i; 2\beta_r)); \gamma/2].$$

Fortunately, this type of non-identifiability does not create any prediction discrepancy, since under GNM, the prediction of y given x is exactly the same for different $(\gamma, \theta_h, \beta, \theta_g)$ and $(\gamma', \theta_h', \beta', \theta_g')$ if we have

$$\gamma^T h(x; \theta_h) = \gamma'^T h(x; \theta_h'), \text{ and } \mathscr{G}^A(\mathbf{x}; \theta_g)\beta = \mathscr{G}^A(\mathbf{x}; \theta_g')\beta'. \quad (4.8)$$

In consideration of the prediction equivalence, a more useful definition of identifiability is given in the following. Let $f(y_i|\mathscr{G}^A(\mathbf{x})_i; \theta_y) = f(y_i|\mathscr{G}^A(\mathbf{x}; \theta_g)_i; \alpha, \beta)$ and $P(r_i = 1|y_i, h(z_i); \theta_r) = P(r_i = 1|y_i, h(z_i; \theta_h); \alpha_r, \gamma, \phi)$, where $\theta_y = (\alpha, \beta, \theta_g)$ and $\theta_r = (\alpha_r, \gamma, \phi, \theta_h)$ contain unknown parameters in the outcome model $Y|\mathbf{x}$ and the missing data model $r|(y, z)$.

The $\mathcal{D}(\theta_y) \otimes \mathcal{D}(\theta_r)$ denotes the domain of (θ_y, θ_r), where $\otimes$ is the tensor product of two spaces.

Definition 4.3.1. *Under GNM, we call* (θ_y, θ_r) *is* ***equivalent*** *to* (θ'_y, θ'_r)*, denoted by*

$$(\theta_y, \theta_r) \sim (\theta'_y, \theta'_r),$$

if (4.8) holds and $\alpha' = \alpha, \alpha'_r = \alpha_r$ *and* $\phi' = \phi$*, where* $\theta_y = (\alpha, \beta, \theta_g), \theta_r = (\alpha_r, \gamma, \phi, \theta_h)$*,* $\theta'_y = (\alpha', \beta', \theta'_g)$*, and* $\theta'_r = (\alpha'_r, \gamma', \phi', \theta'_h)$*. The equivalence class of an element* (θ_y, θ_r) *is denoted by* $[\![(\theta_y, \theta_r)]\!]$*, defined as the set*

$$[\![(\theta_y, \theta_r)]\!] = \{(\theta'_y, \theta'_r) \in \mathcal{D}(\theta_y) \otimes \mathcal{D}(\theta_r) | (\theta'_y, \theta'_r) \sim (\theta_y, \theta_r)\},$$

and the set of all equivalent classes is called the ***Prediction-Equivalent Quotient*** *(PEQ) space, denoted by* $S = \mathcal{D}(\theta_y) \otimes \mathcal{D}(\theta_r) / \sim$ *. The GNM model is called identifiable on the PEQ space iff that*

$$f(y|\mathscr{G}^A(\boldsymbol{x})_i; \theta_y) P(r = 1|y, h(x_i); \theta_r) = f(y|\mathscr{G}^A(\boldsymbol{x})_i; \theta'_y) P(r = 1|y, h(x_i); \theta'_r)$$

holds for all $\boldsymbol{x}, y$ *implies* $(\theta_y, \theta_r) \sim (\theta'_y, \theta'_r)$*.*

Different from identifiability on the parameter space, the identifiability on the PEQ space implies the uniqueness of the prediction given $\mathbf{x}$ instead of parameter estimation. It is applicable to complex architecture that focuses more on prediction than parameter. The following is an example which is not identifiabile on both parameter space and PEQ space.
Example 1. Let $\mathscr{G}^A(x; \theta_g) = x$, $h(x; \theta_h) = x$, $y_i \sim N(\mu + x\beta, 1)$, and $P(r_i = 1|y_i) = [1 + \exp(-\alpha_r - x\gamma - \phi y_i)]^{-1}$ with unknown real-valued $\alpha_r, \gamma, \phi, \mu$ and β, and thus

$$P(r_i = 1|y_i, h(x_i)) f(y_i|\mathscr{G}^A(\mathbf{x})_i) = \frac{\exp[-(y_i - \mu - x_i\beta)^2/2]}{\sqrt{2\pi}[1 + \exp(-\alpha_r - \phi y_i - \gamma x)]}. \tag{4.9}$$

In this case, two different sets of parameters $(\alpha_r, \gamma, \phi, \mu, \beta)$ and $(\alpha_r', \gamma', \phi', \mu', \beta')$ produce equal (4.9) values if $\alpha_r = -(\mu^2 - \mu'^2)/2, \beta' = \beta, \phi = \mu' - \mu, \gamma = \beta(\mu - \mu'), \alpha_r' = -\alpha_r, \phi' = -\phi$, and $\gamma = -\gamma'$. The observed likelihood is only identifiable with ignorable missingness, i.e. $\phi = \phi' = 0$.

Additional conditions are required to ensure the identifiability of GNM on the PEQ space.

Theorem 4.1. *Assume three conditions as follows.*

(A1) For all θ_g, there exist $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ such that $\mathscr{G}^A(\boldsymbol{x}_1; \theta_g)_i \neq \mathscr{G}^A(\boldsymbol{x}_2; \theta_g)_i$ for each i; $\beta \neq 0$ holds.

(A2) For all θ_g and $\boldsymbol{z}$, there exists $(\boldsymbol{u}_1, \boldsymbol{u}_2)$ such that $\mathscr{G}^A([\boldsymbol{z}, \boldsymbol{u}_1]; \theta_g)_i \neq \mathscr{G}^A([\boldsymbol{z}, \boldsymbol{u}_2]; \theta_g)_i$ for each i; and $\beta \neq 0$ holds.

(A3) For all θ_h, there exists (z_1, z_2) such that $h(z_1; \theta_h) \neq h(z_2; \theta_h)$; and $\gamma \neq 0$ holds.

The GNM model (4.2) and (4.5) is identifiable on the PEQ space under Condition (A1). Suppose that there exists an instrumental variable $\boldsymbol{u}$ in $\boldsymbol{x} = [\boldsymbol{z}, \boldsymbol{u}]$ such that $f(y_i | \mathscr{G}^A(\boldsymbol{x})_i)$ depends on $\boldsymbol{u}$, whereas $P(r_i = 1 | y_i, h(x_i))$ does not. Then the GNM model (4.2) and (4.3) is identifiable on the PEQ space under Conditions (A2) and (A3).

Regularity conditions (A1)$\sim$(A3) are easy to satisfy.

4.3.2 Estimation Approach

It is not easy to directly maximize the full likelihood function (4.6) in practice since it can be extremely difficult to compute its integration term. On the other hand, the normality assumption of the error term can be restrictive for GNM consisting of (4.2) and (4.3). Therefore, we propose a doubly robust (DR) estimation approach to alternatively obtain the Inverse Probability Weighted Estimator (IPWE) of θ_y and imputation estimator of θ_r (Robins et al., 1995; Bang and Robins, 2005).

Inverse Probability Weighted Estimator (IPWE) of θ_y

With $\pi(y_i, h(x_i); \theta_r)$ estimated by $\pi(y_i, h(x_i); \widehat{\theta}_r)$, the Inverse Probability Weighted Esti-

mator (IPWE) of θ_y can be obtained by minimizing the weighted cross-entropy loss

$$\mathscr{L}_1(\theta_y|\widehat{\theta}_r) = -\sum_i \frac{r_i}{\pi(y_i, h(x_i); \widehat{\theta}_r)} \sum_{k=1}^{K} 1(y_i = k) \log(P(y_i = k|\mathscr{G}^A(\mathbf{x})_i; \theta_y)) \tag{4.10}$$

when $Y|\mathbf{x}$ follows (4.5) or by minimizing the weighted mean squared error (MSE)

$$\mathscr{L}_1(\theta_y|\widehat{\theta}_r) = \sum_i \frac{r_i}{\pi(y_i, h(x_i); \widehat{\theta}_r)} \{y_i - \alpha - \beta^T \mathscr{G}^A(\mathbf{x}; \theta_g)_i)\}^2 \tag{4.11}$$

when Y is continuous. The estimation equation (4.11) is robust with respect to the normality assumption. If $\pi(y_i, h(x_i); \theta_r)$ is correctly specified, the IPW estimator of θ_y that solves $\partial \mathscr{L}_1(\theta_y|\widehat{\theta}_r)/\partial \theta_y = 0$ is consistent and converges to θ_y according to the following theorem.

Theorem 4.2. *If θ_r is known, then a given estimating function $l(y_i, \mathscr{G}^A(\boldsymbol{x})_i; \theta_y)$ with*

$$E_{\theta_y}\{\sum_i l(y_i, \mathscr{G}^A(\boldsymbol{x})_i; \theta_y)\} = 0$$

satisfies

$$E_{\theta_y}\{\sum_i \frac{r_i}{\pi(y_i, h(x_i); \theta_r)} l(y_i, \mathscr{G}^A(\boldsymbol{x})_i; \theta_y)\} = 0.$$

Imputation estimator of θ_r

With the estimated $f(Y|\mathscr{G}^A(\mathbf{x}; \widehat{\theta}_g))$, we could obtain an estimator of θ_r by minimizing

$$\mathscr{L}_2(\theta_r|\widehat{\theta}_y) = -\sum_{r_i=1} \log(\pi(y_i, h(x_i); \theta_r)) - \sum_{r_i=0} \log(1 - E\{\pi(y_i, h(x_i); \theta_r)|\mathbf{x}; \widehat{\theta}_y\}), \tag{4.12}$$

where $\pi(y_i, h(x_i); \theta_r) = P(r_i = 1|y_i, h(x_i); \theta_r)$ and

$$E\{\pi(y_i, h(x_i))|\mathbf{x}; \widehat{\theta}_y\} = \int P(r_i = 1|y, h(x_i); \theta_r) f(y|\mathscr{G}^A(\mathbf{x})_i; \widehat{\theta}_y) dy.$$

One advantage of our proposed joint estimation approach is that $E(\pi(y_i, h(x_i); \theta_r)|\mathbf{x})$ can be easily approximated by the empirical average of a set of random draws at the nodes with

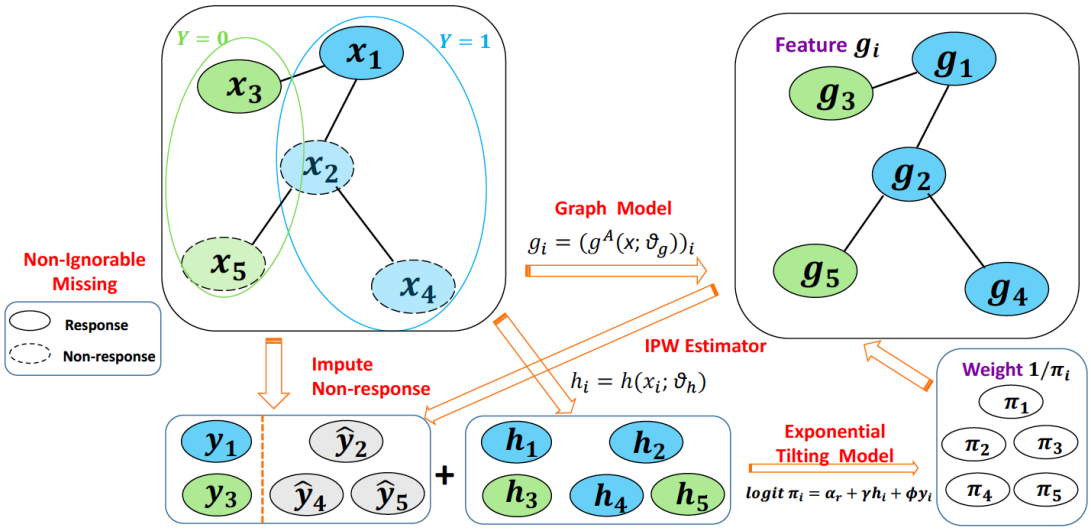

Figure 4.1: General Picture of the Joint Estimation Approach

missing y as the imputed responses:

$$E\{\pi(y_i, h(x_i);\theta_r)|\mathbf{x};\theta_y\} = \int P(r_i = 1|y, h(x_i);\theta_r) f(y|\mathscr{G}^A(\mathbf{x})_i;\theta_y) dy \approx B^{-1}\sum_b \pi(y_{ib}, h(x_i);\theta_r),$$

where $\{y_{ib}\}_{b=1}^B \overset{iid}{\sim} f(y|\mathscr{G}^A(\mathbf{x})_i;\widehat{\theta}_y)$. Thus, we can get an unbiased estimate of (4.12) by replacing the expectation by an empirical mean over samples generated from $f(y|\mathscr{G}^A(\mathbf{x})_i;\widehat{\theta}_y)$ as follows:

$$\widetilde{\mathscr{L}_2}(\theta_r|\widehat{\theta}_y) = -\sum_{r_i=1}\ln(\pi(y_i, h(x_i);\theta_r)) - \sum_{r_i=0}\log(1 - B^{-1}\sum_{y_{ib}\sim f(y|\mathscr{G}^A(\mathbf{x})_i;\widehat{\theta}_y)}\pi(y_{ib}, h(x_i);\theta_r)), \tag{4.13}$$

the gradient of which can be expressed as

$$\nabla_{\theta_r}\widetilde{\mathscr{L}_2}(\theta_r|\widehat{\theta}_y) = -\sum_{r_i=1}\frac{\nabla_{\theta_r}\pi_i}{\pi_i} + \sum_{r_i=0}\frac{B^{-1}\sum_b \nabla_{\theta_r}\pi(y_{ib}, h(x_i);\theta_r)}{1 - B^{-1}\sum_b \pi(y_{ib}, h(x_i);\theta_r)}. \tag{4.14}$$

The imputation estimator of θ_r by minimizing $\mathscr{L}_2(\theta_r|\theta_y)$ is consistent when $f(Y|\mathscr{G}^A(\mathbf{x};\theta_g))$ is correctly specified.The overall estimation procedure is schematically depicted in Figure 4.1.

4.3.3 Algorithm

In this subsection, we provide more details of our proposed imputation and IPW estimation approach about how to jointly estimate θ_y and θ_r by alternatively minimizing the conditional loss functions $\mathscr{L}_1(\theta_y|\widehat{\theta}_r)$ and $\widetilde{\mathscr{L}_2}(\theta_r|\widehat{\theta}_y)$ in practice. Specifically, we update θ_y and then θ_r with $\theta_y^{(e+1)} = \arg\min_{\theta_y} \mathscr{L}_1(\theta_y|\theta_r^{(e)})$ and $\theta_r^{(e+1)} = \arg\min_{\theta_r} \widetilde{\mathscr{L}_2}(\theta_r|\theta_y^{(e+1)})$ in order at each epoch, where $\theta_r^{(e)}$ and $\theta_y^{(e+1)}$ are the estimates of θ_r and θ_y obtained at the e-th and $(e+1)$-th epoch, respectively. We use the gradient descent (GD) algorithm to learn all the parameters in θ_r and θ_y, while incorporating the network architecture of $\mathscr{G}^A(\mathbf{x};\theta_g)$ and $h(x;\theta_h)$.

Without specifying the normal assumption when y_i is continuous, we replace the random draw $y_{ib}^{(e)}$ in (4.13) by the expectation of $\beta_0 + \beta_1^T \mathscr{G}^A(\mathbf{x};\theta_g^{(e)})_i$ at the e-th epoch. It can be seen as an approximation obtained by linearizing $\pi(y_i, h(x_i))$ using a Taylor series expansion and taking the expectation of the first two terms (Beaumont, 2000):

$$E\{\pi(y_i, h(x_i))|\mathbf{x};\theta_y^{(e)}\} \approx \pi(E(y_i|\mathbf{x};\theta_y^{(e)}), h(x_i)) = \pi(\beta_0 + \beta_1^T \mathscr{G}^A(\mathbf{x};\theta_g^{(e)})_i, h(x_i)).$$

In this case, it is equivalent to let $B = 1$ and the sample size, i.e. the total number of nodes will be fixed at each training epoch. Based on simulations and real experiments below, this simplification still outperforms the baseline models with a significant improvement in the prediction accuracy on non-response nodes.

The details of the algorithm are described in five steps as follows:

1. Determine the initial value of the response probability $\pi_i^{(0)}$ (or $\theta_r^{(0)}$). For example, we can let $\pi_i^{(0)} = 1$ for all the labelled vertexes ($r_i = 1$).

2. Let $e = 1$, where e represents the number of epoch. We update θ_y based on $\pi_i^{(0)}$ obtained from the previous epoch by minimizing the loss function in (4.10) using GD. At the i-th iteration within the e-th epoch, we update θ_y as follows:

$$\theta_y^{(e,i+1)} \leftarrow \theta_y^{(e,i)} - \gamma_0 \nabla_{\theta_y} \mathscr{L}_1(\theta_y|\theta_r^{(e-1)}), \tag{4.15}$$

where γ_0 is the learning rate and $\mathscr{L}_1(\theta_y|\theta_r^{(e-1)})$ represents the loss function based on $\pi_i^{(e-1)} = \pi_i(y_i, h(x_i); \theta_r^{(e-1)})$. We denote the updated θ_y as $\theta_y^{(e)}$ after $M^{(e)}$ iterations.

3. Impute y_i for all the unlabelled nodes $r_i = 0$ using $y_i^{(e)} = \beta_0^{(e)} + \mathscr{G}^A(\mathbf{x}; \theta_g^{(e)})_i^T \beta_1^{(e)}$ for the continuous case and sampling $y_i^{(e)}$ from distribution $P(y_i|\mathscr{G}^A(\mathbf{x})_i; \theta_y^{(e)})$ otherwise.

4. We use GD to update θ_r. Specifically, at the j-th iteration, we have

$$\theta_r^{(e,j+1)} \leftarrow \theta_r^{(e,j)} - \gamma_1 \nabla_{\theta_r} \widetilde{\mathscr{L}_2}(\theta_r|\theta_y^{(e)}) \tag{4.16}$$

with the initial start $\theta_r^{(e,0)}$ equal to $\theta_r^{(e-1)}$, and γ_1 is the learning rate. After convergence, we can get the estimate of θ_r denoted as $\theta_r^{(e)}$ at the end of this training epoch. Then we update the sampling weight $\pi_i^{(e)}$ based on $P(r_i = 1|y_i, h(x_i); \theta_r^{(e)})$ for all labelled vertexes.

5. Stop once convergence has been achieved, otherwise let $e = e + 1$ and return to step 3.

The convergence criterion is that whether the imputed unlabelled vertexes at epoch e only slightly differ from those at epoch $(e-1)$. In other words, the iteration procedure is stopped if

$$\sum_{r_i=0} |y_i^{(e)} - y_i^{(e-1)}| / \sum_i 1(r_i = 0) \leq \varepsilon$$

We let M_0 and M_1 be the maximal number of allowed internal iterations at each epoch for updating θ_y and θ_r, respectively. For more details, you can refer to the Algorithm 1.

4.4 Experiments

In this section, simulations and one real data analysis are conducted to evaluate the empirical performance of our proposed methods and a baseline method, which ignores the non-response (SM). In the real data part, GNM is also compared with the model with a misspecified ignorable missing mechanism, and some other state-of-art 'de-biasing' methods. In the simulation part, we simulate the node response y based on (4.3) and generate the labelled set by the exponential tilting model (4.2). For the real data analysis, we evaluate all

the compared models by a semi-supervised document classification on the citation network-Cora with non-ignorable non-response.

Algorithm 1 Gradient Descent-based Joint Estimation Procedure

Input: $\mathbf{x} \in R^{N\times p}$; $r_i y_i$ for $\forall i$; $A \in R^{N\times N}$

Initialize $\pi_i^{(0)}, \theta_r^{(0)}, \theta_y^{(0,0)}$; $e = 0$
while $\sum_{r_i=0} |y_i^{(e)} - y_i^{(e-1)}| / \sum_i 1(r_i = 0) > \varepsilon$ **do**
  $e \leftarrow e + 1$; $w_0, w_1 = 0$; $\mathscr{L}_1(\theta_y^{best}|\theta_r^{(e-1)}), \widetilde{\mathscr{L}_2}(\theta_r^{best}|\theta_y^{(e)}) = \infty$
  for $i \leftarrow 0$ to $(M_0 - 1)$ **do**
    $\theta_y^{(e,i+1)} \leftarrow \theta_y^{(e,i)} - \gamma_0 \nabla_{\theta_y} \mathscr{L}_1(\theta_y|\theta_r^{(e-1)})$
    if $\mathscr{L}_1(\theta_y^{(e,i+1)}|\theta_r^{(e-1)}) < \mathscr{L}_1(\theta_y^{best}|\theta_r^{(e-1)})$ **then**
      $\theta_y^{best} \leftarrow \theta_y^{(e,i+1)}$
    else
      $w_0 \leftarrow w_0 + 1$
      if $w_0 > P_0$ **then**
        break
      end if
    end if
  end for
  $\theta_y^e \leftarrow \theta_y^{(e,i)}$
  for $j \leftarrow 0$ to $(M_1 - 1)$ **do**
    $\theta_r^{(e,j+1)} \leftarrow \theta_r^{(e,j)} - \gamma_1 \nabla_{\theta_r} \widetilde{\mathscr{L}_2}(\theta_r|\theta_y^{(e)})$
    if $\widetilde{\mathscr{L}_2}(\theta_r^{(e,j+1)}|\theta_y^{(e)}) < \widetilde{\mathscr{L}_2}(\theta_r^{best}|\theta_y^{(e)})$ **then**
      $\theta_r^{best} \leftarrow \theta_r^{(e,j+1)}$
    else
      $w_1 \leftarrow w_1 + 1$
      if $w_1 > P_1$ **then**
        break
      end if
    end if
  end for
  $\theta_r^e \leftarrow \theta_r^{(e,j)}$
end while

In this dissertation, we use GCN (Kipf and Welling, 2016) to learn the latent node representations $\mathscr{G}^A(\mathbf{x})$ with the layer-wise propagation defined as

$$H^{(l+1)} = f(H^{(l)}, A) = \sigma(\widehat{D}^{-\frac{1}{2}}\widehat{A}\widehat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}), \tag{4.17}$$

where $\widehat{A} = A + I$, in which I is an identity matrix, and $\widehat{D}$ is the diagonal vertex degree matrix of $\widehat{A}$. The $W^{(l)}$ is a weight matrix for the l-th layer and $\sigma(\cdot)$ is an non-linear activation function. $H^{(0)} = \mathbf{x}$ is the initial input and $\mathscr{G}^A(\mathbf{x}) = H^{(2)} \in R^{N \times \bar{p}}$ is the output of the second layer-wise propagation. To be fair, we let $\mathscr{G}^A(\mathbf{x})$ be a 2-layer GCN model for all compared approaches.

4.4.1 Simulations

We consider a network data generated by $|V| = 2708$ vertexes together with a binary adjacency matrix A. $\mathbf{x} \in R^{2708 \times 1433}$ denotes the fully observed input features which is a large-scale sparse matrix. Both A and $\mathbf{x}$ are obtained from the Cora dataset. The node response is simulated from the following model:

$$y_i = \beta_0 + \beta_1^T \mathscr{G}^A(\mathbf{x})_i + \epsilon_i, \tag{4.18}$$

where $\epsilon_i \sim N(0, \sigma^2)$ and $\mathscr{G}^A(\mathbf{x})$ is the output of a 2-layer GCN model. We let response probability π depend on the unobserved vertex response y only , and (4.2) is simplified to

$$\pi_i \equiv P(r_i = 1|y_i) = \frac{\exp\{\alpha_r + \phi y_i\}}{1 + \exp\{\alpha_r + \phi y_i\}}. \tag{4.19}$$

In this case, the instrumental variable $\mathbf{u}$ is exactly $\mathbf{x}$ itself, and the identifiability automatically holds according to Theorem 4.1. All β's in (4.18) are sampled from uniform distribution $U(0, 1)$. The α_r and ϕ were selected to make the overall missing proportion be approximately 90%. The labelled subset are randomly split into training and validation sets, while the remaining non-response nodes build the testing set. We train all the compared models for a maximum of 200 epochs ($E = 200$) using Adam (Kingma and Ba, 2014) with a learning rate 0.05 and make predictions $\widehat{y}_i$ for each testing vertex. Training is stopped when validation loss does not decrease in 15 consecutive iterations. We keep all other model settings used by (Kipf and Welling, 2016) and fix the unit size of the first hidden layer to be 16.

Table 4.1 summarizes the estimation results under different $(\bar{p}, \sigma)$ combinations, where

root mean squared error (RMSE) and Mean absolute percentage error (MAPE) are computed between the true node response y and prediction $\widehat{y}$ over the 50 runs. We can clearly see that GNM outperforms SM under all the four settings with much smaller mean RMSEs and MAPEs. Moreover, GNM is more stable than SM with smaller estimation variance.

$\bar{p}$	σ	Method	Metric	Mean	SD
4	0.5	SM	RMSE	1.1925	6.43e-1
			MAPE	0.2932	2.01e-1
		GNM	RMSE	0.6983	1.28e-2
			MAPE	0.1995	1.00e-2
	1	SM	RMSE	1.6185	8.58e-2
			MAPE	0.3104	4.73e-2
		GNM	RMSE	1.2103	4.81e-2
			MAPE	0.2263	2.28e-2
16	0.5	SM	RMSE	0.7923	9.94e-2
			MAPE	0.2014	2.42e-2
		GNM	RMSE	0.6015	2.17e-2
			MAPE	0.1672	1.90e-2
	1	SM	RMSE	1.4212	2.14e-1
			MAPE	0.2129	1.05e-2
		GNM	RMSE	1.1316	6.04e-2
			MAPE	0.1849	4.62e-3

Table 4.1: Mean RMSEs and MAPEs by GNM and SM based on simulated data sets

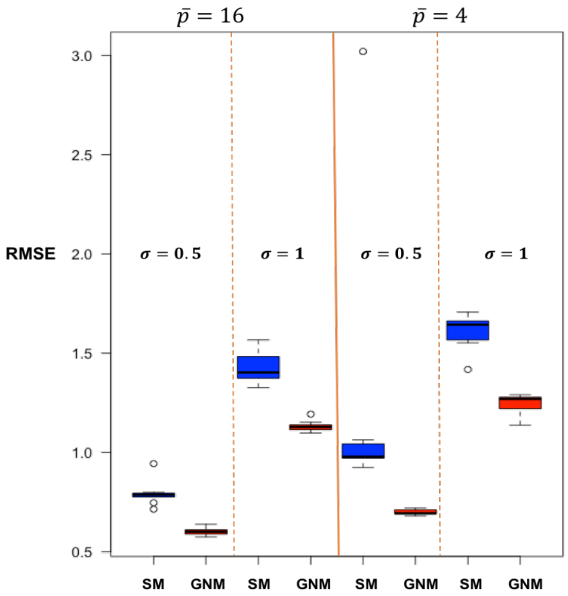

Figure 4.2: Boxplot of RMSEs in real data analysis

4.4.2 Real Data Analysis

For the real data analysis, we modify the Cora to a binary-class data by merging the six non-'Neural Network' classes together. The global prevalence of two new classes are $(0.698, 0.302)$ with $N_0 = \#\{y = 0\} = 1890$ and $N_1 = \#\{y = 1\} = 818$, respectively.

Two missing mechanisms are considered. A simple setup is the same as (4.19). In this case, we compare our method with the inverse weighting approach proposed by Rosset et al. (2005). We let the two functions of x required to estimate π under their framework to be the constant 1 and the first principle component (PC) score, which is more stable compared to other functions such as a general x_j or $\sum_j x_j$. In a more complicated setup, the labelled

nodes are generated based on

$$\pi_i \equiv P(r_i = 1|y_i, h(x_i)) = \frac{\exp\{\alpha_r + \gamma^T h(x_i) + \phi y_i\}}{1 + \exp\{\alpha_r + \gamma^T h(x_i) + \phi y_i\}}, \tag{4.20}$$

where $h(x_i) = \exp(\sum_j x_{ij}/a_0 - a_1) - (\sum_j x_{ij} - a_2)/a_3$ with value range being $[0, 1]$. The explicit form of $h(x)$ is assumed to be unknown and we use a multi-layer perceptron to approximate it. The network has two hidden layers with 128 and 64 units. respectively, and we use the 'tanh' activation for the final output layer. As a comparison, we also include the results when the 'non-ignorable' missingness is over-simplified to the 'ignorable' one (GIM). We let $n_k = \#\{(y_i = k) \wedge (r_i = 1)\}$, and use λ to denotes the size ratio between the two groups of labelled nodes, i.e. n_1/n_0.

		Accuracy	
λ	Method	Mean	SD
1	SM	0.8683	1.98e-2
	Rosset	0.8514	5.19e-2
	GNM	**0.8947**	**6.47e-3**
1.5	SM	0.8458	2.21e-2
	Rosset	0.8311	7.09e-2
	GNM	**0.8908**	**1.26e-2**
2	SM	0.8052	3.26e-2
	Rosset	0.8193	6.05e-2
	GNM	**0.8648**	**2.54e-2**

Table 4.2: Mean Prediction Accuracy for the simple setup by each method

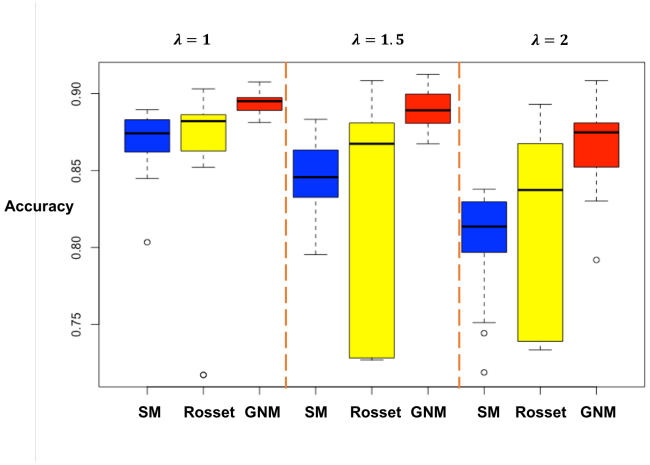

Figure 4.3: Boxplot Prediction Accuracy for the simple setup

Results are summarized in Tables 4.2 and 4.3. Reported values represent the average classification accuracy on testing data by 50 replications with re-sampling allowed. In each setup, two 'de-biasing' methods including our approach are compared with SM. We adjust α and β to make the size of training set be around 120 for each sub-setting. Increasing λ reduces the number of included $y = 0$ nodes in the training set, leading to an insufficient learning power and thus a lower overall classification accuracy. For the simple setup, GNM significantly outperforms compared models by increasing the baseline prediction accuracy

		Accuracy	
λ	Method	Mean	SD
1	SM	0.8663	1.21e-2
	GIM	0.8713	1.52e-2
	GNM	**0.8961**	**1.18e-2**
2	SM	0.8141	2.34e-2
	GIM	0.8291	2.79e-2
	GNM	**0.8669**	**1.63e-2**

Table 4.3: Mean Prediction Accuracy for the complicated setup by each method

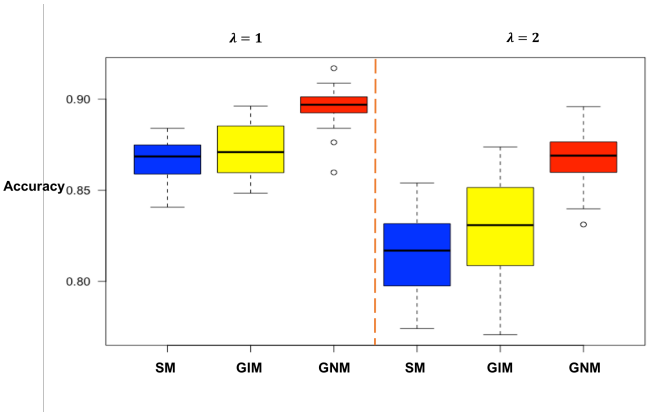

Figure 4.4: Boxplot of Prediction Accuracy for the complicated setup

by 3.1% - 7.4%. On the other hand, GNM is less sensitive to the sample selection and has smaller variance compared to the method by Rosset et al. (2005). For the complicated setup, mis-specifying the 'Non-Ignorable' missingness as 'Ignorable' still has big biases even though achieving some improvement against SM. The mean prediction accuracy by GNM is between 3.7% to 4.8% higher than that by GIM.

In both sub-settings, our method always leads to the smallest estimation variance, which is less affected by the selection of labelled nodes. For both setups, higher λ value leads to bigger sampling bias, and subsequently there is more significant improvement in the prediction accuracy. Figures 4.3 and 4.4 are the boxplots of prediction accuracy obtained from each method under the two model setups. It may intuitively demonstrates the necessity of taking into account missing mechanism in order to achieve higher prediction accuracy on the unlabelled nodes.

CHAPTER 5: STOD: SPATIAL-TEMPORAL ORIGIN -DESTINATION PREDICTION MODEL

5.1 Introduction

Our aim is to introduce a hierarchical Spatial-Temporal Origin-Destination (STOD) prediction model to jointly extract the complex spatial-temporal features of OD data by using some well-designed CNN-based architectures. Instead of modelling the dynamic OD networks as a sequence of images and applying standard convolution filters to capture their spatial information, we introduce a novel Spatial Adjacent Convolution Network (SACN) that uses irregular convolution filters to cover the most related OD flows for a target one. The OD flows connected by common starting and/or ending vertexes, which may fall into different regions in O_t, can be spatially correlated and topologically connected. Moreover, for most ride-sharing platforms, a passenger is more likely to send a new request from the location where his last trip ends in. Thus, to learn such sequential dependency, we introduce a temporal gated CNN (TGCNN) (Yu et al., 2018) and integrate it with SACN by using the sandwich-structured ST-conv block in order to collectively catch the evolutionary mechanism of dynamic OD flow systems. A periodically shifted attention mechanism is used to capture the shift in the long-term temporal periodicity. Then, the combined short-term and long-term spatial-temporal representations are fed into the final prediction layer to complete the whole architecture.

To examine the prediction performance of our STOD model, we use a large-scale customer request data with available OD coordinates obtained from a large ride-sharing platform. The dataset contains three-month platform orders in the city of Beijing, where $N = 50$ locations are selected and in total $N^2 = 2500$ OD flows are generated within every 30 minutes, valued by the demands amount between each pair of vertexes. We compare our STOD model with

many state-of-art methods in predicting the OD flows of customer requests. Some methods are traditional ones, whereas others are based on deep learning.

The main contributions are summarized as follows:

- We propose a latent deep learning model for OD flow prediction problems, which automatically extracts the spatial-temporal features of OD flow data.
- We design a novel SACN to capture the semantic connections and functional similarities among correlated OD flows, by modelling each flow network snapshot as a graph adjacency matrix.
- We use CNN-based architectures to learn the temporal dependency and use the periodically shift attention mechanism to capture the shift of the long-term periodicity.
- Experimental results on a real customer demand data set obtained from a ride-sharing platform demonstrate that STDO outperforms many state-of-art methods in OD flow prediction, with 6.5% and 7.3% improvement of testing RMSE.

5.2 Definitions and Problem Statement

For a given urban area, we observe a sequence of adjacency matrices representing the OD flow maps defined on a fixed vertex set V, which indicates the N selected sub-regions from this area. We let $V = \{v_1, v_2, \ldots, v_N\}$ denote the vertex set with v_i being the i-th sub-region. The shape of each grid v_i could be either rectangles, hexagons or irregular sub-regions. We define the dynamic OD flow maps as $\{O_{1,1}, \ldots, O_{1,T}, \ldots, O_{D,1}, \ldots, O_{D,T}\}$, where $d \in \{1, \ldots, D\}$ and $t \in \{1, \ldots, T\}$ represent the day and time indexes, respectively. For each snapshot $O_{d,t}$, the edge weight $o_{d,t}^{ij}$ at row i column j denotes the flow amount from node v_i to node v_j at time t of day d. A larger edge weight $o_{d,t}^{ij}$ is equivalent to a strong connection between vertexes v_i and v_j. The $O_{d,t}$s' are asymmetric since all the included OD flows are directed. Specifically, we have $o_{d,t}^{ij} = 0$ if there is no demand from v_i to v_j within the t-th time interval of day d.

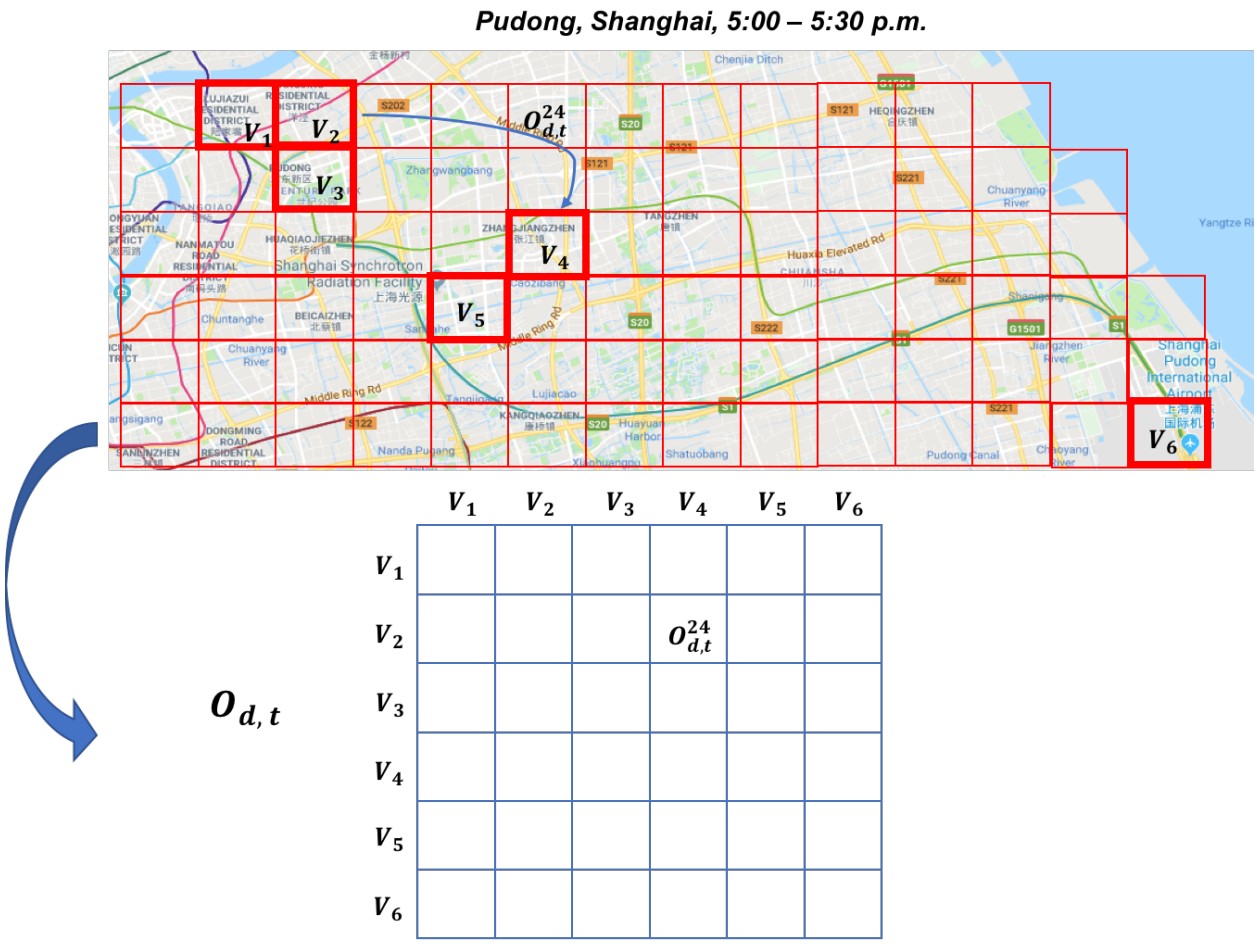

Figure 5.1: A real example of customer demands from ride-sharing platforms to explain OD flow data from the perspective of dynamic graph adjacency matrices

We introduce a motivating example to more clearly understand above definitions. Figure 5.1 presents one snapshot of OD flow networks acquired from a real-world customer requests data set. We divide the Pudong area in the city of Shanghai into many non-overlapping square grids, from which v_1 to v_6 are picked out to build the vertex set V as the upper sub-figure demonstrates. The plotted timestamp covers a time range from 5:00 p.m. to 5:30 p.m., and the corresponding adjacency matrix $O_{d,t}$ in the lower sub-figure include all the 6^2 OD flows. The element in row i, column j denotes the total number of customer requests received by the ride-sharing platform within this 30 minutes from an origin node v_i to the destination one v_j.

The goal of the prediction problem is to predict the snapshot $O_{d,t+j} \in R^{N\times N}$ in the future time window $(t+j)$ of day d given previously observed data, including both short-term and long-term historical information. The short-term data consists of the last p_1 timestamps from $t+1-p_1$ to t, denoted by $\mathbf{O}_1 = \{O_{d,t+1-p_1}, O_{d,t+1-p_1+1}, \ldots, O_{d,t}\}$. The long-term data is made up of q time series $\{O_{d-\varphi,t+j-(p_2-1)/2}, \ldots, O_{d-\varphi,t+j+(p_2-1)/2}\}$ of length p_2 for each previous day $(d-\varphi)$, where $\varphi = 1, \ldots, q$, with the predicted time index $(t+j)$ in the middle. We let $\mathbf{O}_2 = \{O_{d-q,t+j-(p_2-1)/2}, \ldots, O_{d-q,t+j+(p_2-1)/2}, \ldots, O_{d-1,t+j-(p_2-1)/2}, \ldots, O_{d-1,t+j+(p_2-1)/2}\}$ denote the entire long-term data. Increasing p_1 and p_2 leads to the training context size, and subsequently higher prediction accuracy, but more training time.

We reformulate the sequence of short-term OD networks $\mathbf{O}_1$ into a 4D tensor $O_{ST} \in R^{N\times N\times p_1\times 1}$ and concatenate the long-term snapshots $\mathbf{O}_2$ into a 5D tensor $O_{LT} \in R^{q\times N\times N\times p_2\times 1}$. The ST and LT here stand for 'short-term' and 'long-term', respectively. We can formally define the final prediction problem by using both short-term and long-term historical information as follows:

$$o_{d,t+j} = F(O_{ST}, O_{LT}), \tag{5.1}$$

where $F(\cdot,\cdot)$ represents the STOD model, which captures the network structures of OD flow data as well as the temporal dependencies in multiple time scales.

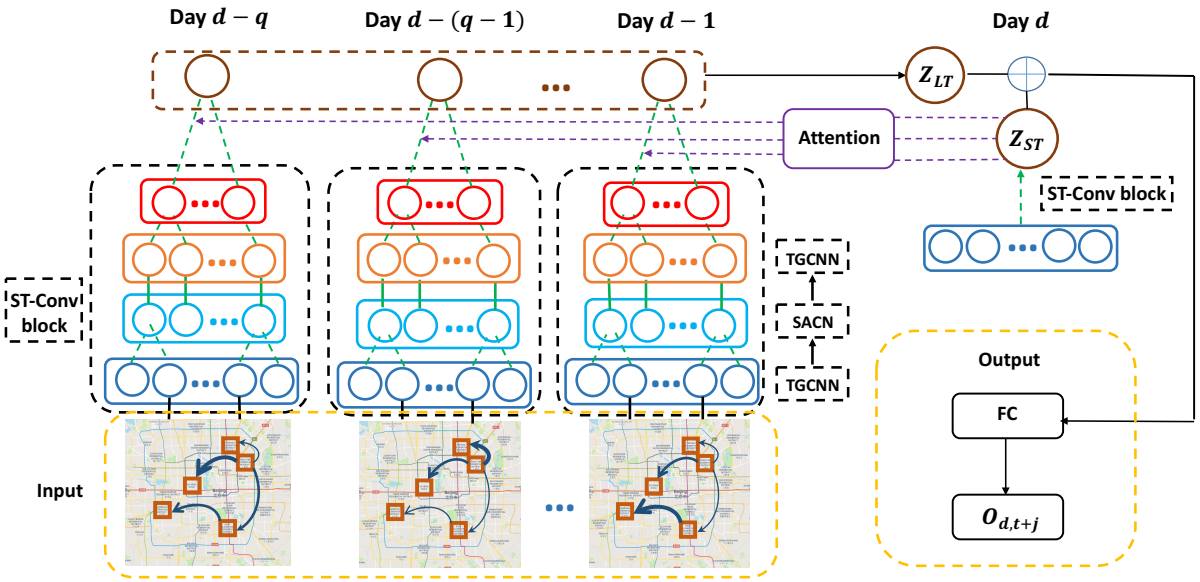

Figure 5.2: The Architecture of STOD model

5.3 STOD Framework

In this section, we describe the details of our proposed Spatial-Temporal Origin-Destination (STOD) prediction model. It consists of three main components: a novel CNN-based SACN, a temporal gated CNN (Yu et al., 2018), and a modified periodically shifted attention mechanism. First, we introduce SACN using irregular CNN filters to capture the spatial features of network snapshot at each timestamp t, which accounts for the relationships among neighboring OD flows on the weighted graph structure. Second, we use gated CNN to learn the temporal dependency, which is computationally efficient especially for a long time sequence, while achieving competitive results with LSTM. We use a sandwich-structure ST-Conv block to jointly capture the evolving patterns of dynamic OD flow maps. Moreover, we modify the periodically shifted attention mechanism proposed by Yao et al. (2018) to catch the shifting of the long-term periodicity by measuring the similarity between the short-term and long-term representations. Figure 5.2 shows the architecture of STOD model. In the rest of this section, we will discuss the details of these main structures of STOD model in order.

5.3.1 Spatial Adjacent Convolution Network

As we mentioned above, directly applying standard CNN operations to the dynamic OD flow map $O_{d,t}$ disregards the connections between neighboring OD flows in the network. For a target OD flow $o_{d,t}^{ij}$, the nearby OD flows in $O_{d,t}$, such as $o_{d,t}^{kl}$, may be unrelated from the perspective of graph. Let's consider the 3×3 receptive filed with $o_{d,t}^{ij}$ in the center by a standard CNN filter. The upper-left, upper-right, lower-left and lower-right OD flows in the current kernel window provide less information compared to those OD flows out of the 3×3 region but sharing common nodes with $o_{d,t}^{ij}$. Moreover, if we change the order of the N vertexes in $O_{d,t}$, then the network structure is unchanged, but a different set of OD flows will be covered by the 3×3 receptive filed with the central element being $o_{d,t}^{ij}$. Figure 5.3 illustrates why standard CNN cannot capture enough network information by using a real-world example.

Figure 5.3 depicts the same snapshot of demand flow maps as Figure 5.1 from a ride-sharing platform. For the OD flows starting from Lu Jia Zui, the central business district of Shanghai, to Pudong airport, as illustrated in the upper sub-figure, the most related OD flows should be those with either origin or destination being Pudong airport or Lu Jia Zui within the past few timestamps. It is reasonable to assume that someone from Zhang Jiang, the high-tech park of the Pudong district, finishing attending a business meeting at Lu Jia Zui, may need a ride to the Pudong airport for leaving. Therefore, a certain part of the travel requests from Lu Jia Zui to Pudong airport in the current time window can be matched with some historical finished trips from a third-party location to Lu Jia Zui by the same group of passengers. However, as the lower-left sub-figure illustrates, some of the OD flows covered by a single CNN filter (the green square) are not significantly correlated with the central flows from Lu Jia Zui to Pudong airport. The four OD flows in the corners of the kernel window do not share origin or destination nodes with the central OD flow, and thus they may be topologically far away from the target one in the graph.

As the lower right sub-figure shows, OD flows with either origin or destination being v_i

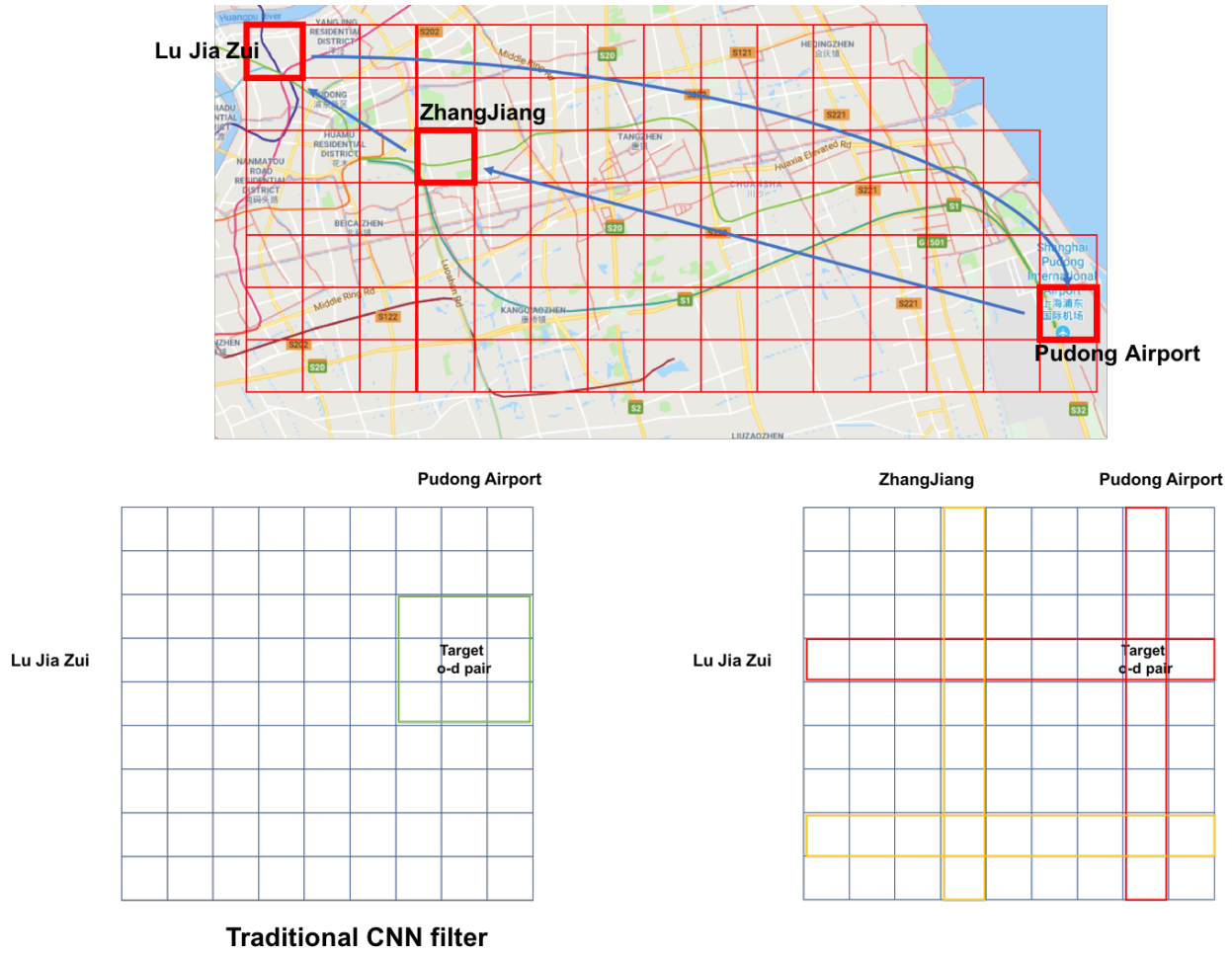

Figure 5.3: An empirical example of passenger requests to illustrate how standard CNN fails to capture the network structure of OD flow data

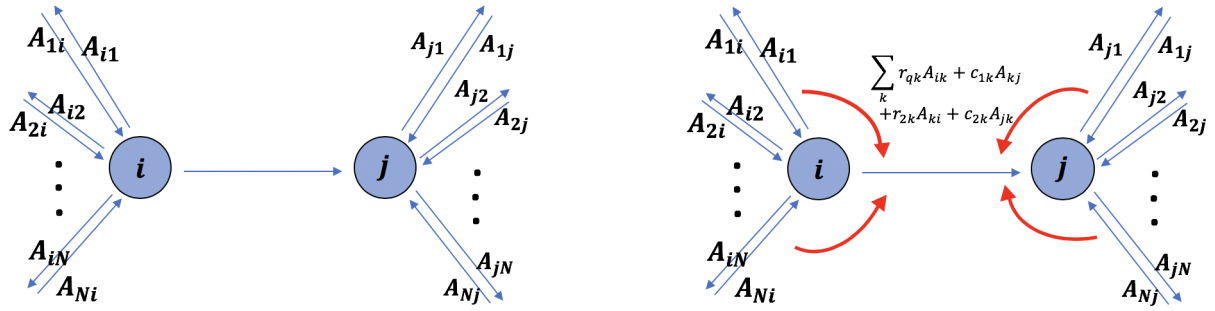

Figure 5.4: Working mechanism of spatial adjacent convolution network (SACN) for a target OD flow from v_i to v_j

or v_j, covered by the red and yellow kernel windows, are considered to be the most related ones for $o_{d,t}^{ij}$ in row i and column j. Kawahara et al. (2017) introduces a novel edge-to-edge convolutional operator that leverages the topological locality of graph adjacency matrices. Different from standard CNN filters that pay attention to spatially nearby pixels on the image space, the edge-to-edge layer utilizes a novel receptive field to cover elements in the same row or column (the red window) with the target OD flow. However, the OD flows with destination being v_i or origin being v_j (the yellow window) may be more semantically correlated according to the real example we discussed above. A new trip starting from v_i is very likely to follow an old one ending at v_i by the same customer.

We propose a novel CNN-based architecture SACN using a global-view receptive field to include all connected edges in the graph and exclude the topologically unrelated ones. Formally, we use SACN to extract the latent topological structure inside the OD flow network $O_{d,t}$ at each timestamp (d, t). For an L-layer SACN architecture, the l-th layer takes M^{l-1} edge features obtained from the previous $(l-1)$-th layer as input and feeds the M^l-dimensional output to the next layer. The input of a general SACN layer l is a 3D tensor, $A_{d,t}^l \in R^{N \times N \times M^l}$, which includes the M^l features of each of the N^2 OD flows, and the output is another 3D tensor $A_{d,t}^{l+1}$ of size $N \times N \times M^{l+1}$. As illustrated in Figure 5.4, the learned representation of a target edge is defined as the weighted sum of those from the same row or column in the adjacency matrix, and those from the row or column in the transposed adjacency matrix.

The output of the l-th layer-wise SACN propagation for the OD flow from v_i to v_j, denoted as $A_{d,t}^{ij,n}(l+1)$, is written as

$$\mathcal{F}\{\sum_{m=1}^{M^l}\sum_{k=1}^{N} r_1^{km,n}(l)A_{d,t}^{ik,m}(l) + c_1^{km,n}(l)A_{d,t}^{kj,m}(l) + r_2^{km,n}(l)A_{d,t}^{ki,m}(l) + c_2^{km,n}(l)A_{d,t}^{jk,m}(l)\} \quad (5.2)$$

where $A_{d,t}^{ij,n}(l+1)$ denotes the n-th output feature by the l-th SACN layer for the OD flow $o_{d,t}^{ij}$ in row i, column j of graph snapshot $O_{d,t}$ for $n = 1, \ldots, M^l$. The $\{r_1^{km,n}(l)\}, \{r_2^{km,n}(l)\}, \{c_1^{km,n}(l)\}$, $\{c_2^{km,n}(l)\} \in R^{N\times M^l \times M^{l+1}}$ include all the related parameters to be learnt for the l-th SACN layer. The $\mathcal{F}(\cdot)$ represents an elementwise activation function, such as ReLU(x) $= \max(0, x)$. The first part of (5.2) works by summing up the feature values of OD flows having either the same origin or destination with the target OD flow. The second part covers another set of OD flows that either start at v_j or end at v_i. Therefore, the receptive field of SACN includes the two rows and two columns colored by red and yellow as demonstrated by the lower-right sub-figure in Figure 5.3. Similar to standard CNN architectures, OD flows more related to the target one are more highly weighted by a multi-layer SACN operator.

For an L-layer SACN model, the output at the final L-th layer, denoted as $A_{d,t}^{ij,n}(L+1)$, is defined as follows:

$$\mathcal{F}\{\sum_{m=1}^{M^L}\sum_{k=1}^{N} r_1^{km,n}(L)A_{d,t}^{ik,m}(L) + c_1^{km,n}(L)A_{d,t}^{kj,m}(L) + r_2^{km,n}(L)A_{d,t}^{ki,m}(L) + c_2^{km,n}(L)A_{d,t}^{jk,m}(L)\}, \quad (5.3)$$

where $A_{d,t}^{ij,n}(L+1)$ is the n-th feature map of the final output. Then, the overall spatial representations captured by an L-layer SACN can be defined as $s_{d,t} = [A_{d,t}^{L+1,1}, A_{d,t}^{L+1,2}, \ldots, A_{d,t}^{L+1,M^{L+1}}] \in R^{N\times N\times M^{L+1}}$. For notational simplification, we use $A(\theta)*_L$ to represent a L-layer SACN operator, where θ includes all parameters to be learnt.

5.3.2 Temporal Gated CNNs

Canonical recurrent networks, such as LSTMs, have been widely used to model temporal dependency by maintaining a hidden activation that is propagated through time. These

approaches suffer from the problem of lower training efficiency, gradient instability, and time-consuming convergence. The high dimension of the spatial representations $s_{d,t}$ captured by SACN and a potential long temporal sequence length make RNN architectures notoriously difficult to train. Recent studies pay more attention to convolutional architectures for modelling sequential data. Yu et al. (2018) introduced a CNN-based operator with gate mechanism to learn the intrinsically sequential dependency. The pure convolutional architecture is more flexible in handling various data structures and the gate mechanism decides the relevant information to be passed through. Yu et al. (2018) pointed out that this special design allows parallel and controllable training procedures to increase convergence speed, A hierarchical feature maps could be generated through a multiple-filter architecture.

The temporal gated CNN (TGCNN) consists of two parts including one being a 3D convolution kernel applied to the spatial representations of all the N^2 OD flows along the time axis and the other being a gated linear units (GLU) as the gate mechanism. Given the spatial feature maps of m_0 channels or the original OD flow data ($m_0 = 1$) at each of r successive time intervals, we can generate a 4D tensor of size $N \times N \times r \times m_0$.

The temporal gated CNN uses a 3D convolutional kernel of size $1 \times 1 \times K$ with zero padding. Applying the filter each single time shortens the sequence length by $(K - 1)$ with the first two dimensions of the input array, which correspond to the total number of OD flows unchanged. The output at each position in the new sequence would be the weighted sum of K mapped points in the input sequence. Thus, $2m_1$ temporal gated CNN filters map a r-length spatial-temporal sequential data $y \in R^{N \times N \times r \times m_0}$ in feature depth m_0 to a new sequence $[P\ Q] \in R^{N \times N \times (r-K+1) \times (2m_1)}$ of length $(r - K + 1)$. The P and Q, in the same size with m_1 channels, serve as the learned temporal representations and the selection gate, respectively. Thus, the detailed architecture of a one-layer temporal gated CNN is formally defined as follows:

$$G(\gamma) *_\tau y = P \odot \sigma(Q) \in R^{N \times N \times (r-K+1) \times m_1}, \tag{5.4}$$

where $\odot$ denotes the element-wise Hadamard product and γ denotes the set of parameters

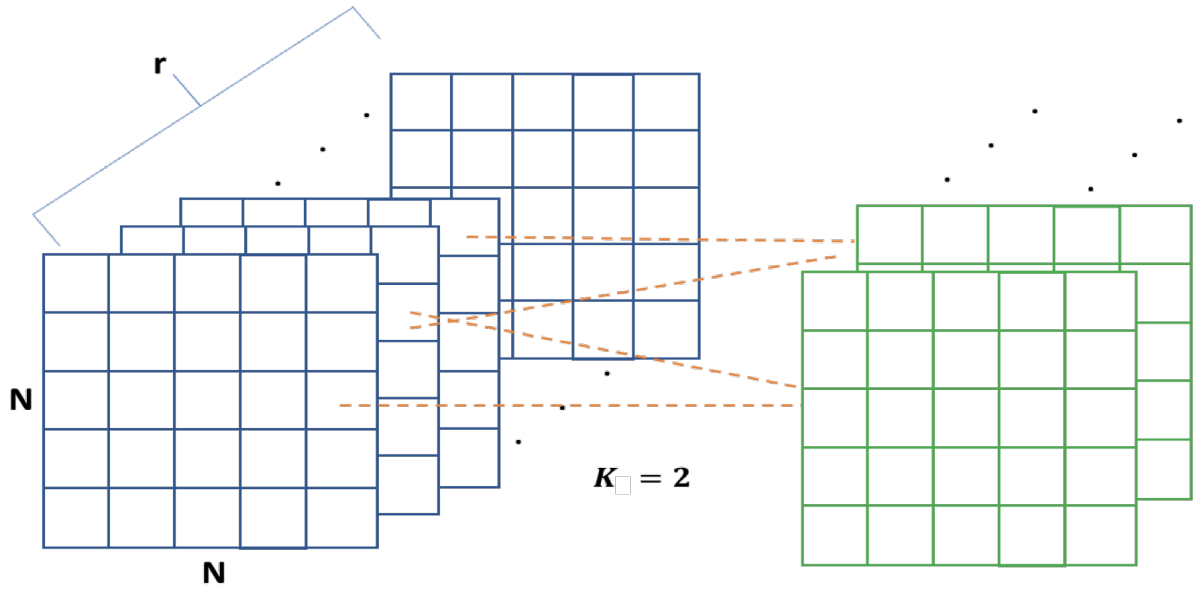

Figure 5.5: Illustration of temporal gated CNN with kernel size being $1 \times 1 \times 2$ in capturing temporal dependency and reducing sequence length

to be learnt. The output Q with an element-wise sigmoid function $\sigma(\cdot)$ work together as a gate mechanism to evaluate the importance of each element in P and assign a weight before being passed to the following layer. A simple graphical example is described in Figure 5.5 to illustrate how the temporal gated CNN works for modeling the temporal dependency of the OD flow data.

5.3.3 ST-Conv blocks

Motivated by (Yu et al., 2018), we build a spatial-temporal convolutional block (ST-conv block) to jointly capture the spatial-temporal features of OD flow data by combining the proposed SACN with TGCNN. The ST-Conv block has a 'sandwich'-structure architecture with an L-layer SACN operator in the middle connecting the two TGCNN layers on both sides. Based on the experiment results of (Yu et al., 2018), we shall conclude that the 'sandwich' structure can not only jointly capture spatial-temporal representations of the OD flow data, but also dynamically shorten the sequence length of the input data to dramatically reduce the training load that the memory needs when SACN extracts spatial patterns.

Both the input and output of a single ST-Conv block are 4D tensors. We let the spatial-

temporal representation $y^0 \in R^{N\times N\times r\times c_0}$ of c_0 features be the input, which can be the original OD flow data by setting $c_0 = 1$. The mathematical definition of the ST-Conv block is defined as

$$y^1 = G_1(\gamma_1) *_\tau [A(\theta_0) *_L \{G_0(\gamma_0) *_\tau y^0\}], \tag{5.5}$$

where $G_1(\cdot)$ and $G_0(\cdot)$ are the two temporal gated CNN layers and $A(\theta)*_L$ is an L-layer SACN operator. The $(\theta_0, \gamma_0, \gamma_1)$ is the set of all parameters to be learnt. The m_1 3D convoluitonal filters of kernel size $1 \times 1 \times K_0$ and $1 \times 1 \times K_1$ are used by the two TGCNN $G_0(\gamma_0)*_\tau$ and $G_1(\gamma_1)*_\tau$, respectively. The L-layer SACN is applied to each 3D snapshot of size $N \times N \times m_1$ obtained from TGCNN $G_0(\gamma_0)*_\tau$, and then fed into the other TGCNN operator $G_1(\gamma_1)*_\tau$. One ST-Conv block shortens the temporal length of input y^0 by $(K_0 + K_1 - 2)$, and the dimension of the output y^1 becomes $N \times N \times \{r - (K_0 + K_1 - 2)\} \times m_1$. Accordingly, a set of $n_{ST} = (r - 1)/(K_0 + K_1 - 2)$ ST-Conv blocks reduces the sequential length from r to 1. We can then flatten the spatial-temporal representation into a 3D tensor of size $N \times N \times m_1$ by squeezing out the temporal dimension.

The short-term spatial-temporal representation $z_{ST} \in R^{N\times N\times c_{ST}}$ is obtained by continuously applying $(p_1 - 1)/(K^0_{ST} + K^1_{ST} - 2)$ ST-Conv blocks to the short-term OD flow data $O_{ST} \in R^{N\times N\times p_1\times 1}$. The kernel sizes of the two TGCNNs in all ST-Conv blocks are fixed to be $1 \times 1 \times K^0_{ST}$ and $1 \times 1 \times K^1_{ST}$, respectively. The c_{ST} filters are used by both the L-layer SACN and the two TGCNNs. The detailed propagation of the n-th ST-Conv block is defined as

$$z^{n+1}_{ST} = G_1(\gamma^1_{ST}) *_\tau [A(\theta_{ST}) *_L \{G_0(\gamma^0_{ST}) *_\tau z^n_{ST}\}], \tag{5.6}$$

where z^n_{ST} is the input obtained from the $(n-1)$-th ST-Conv block and z^{n+1}_{ST} is the output, which will then be fed into the following $(n+1)$-th ST-Conv block. The $(\theta_{ST}, \gamma^0_{ST}, \gamma^1_{ST})$ contains all the related parameters. Specifically, the initial input for the 1-st ST-Conv block z^1_{ST} is the original OD flow data $O_{ST} \in R^{N\times N\times p_1\times 1}$. The $z_{ST} = z^{n_{ST}+1}_{ST} \in R^{N\times N\times c_{ST}}$ is the output of the last n_{ST}-th ST-Cov block.

5.3.4 Periodically Shifted Attention Mechanism

In addition to capturing the the spatial-temporal features from short-term OD flow data O_{ST}, we also take into account the long-term temporal periodicity since there exists some day-wise cycling characteristic hidden in the OD flow data, which is caused by customer's travelling schedule and the city's traffic pattern. Looking back through a big time scope by directly applying ST-Con blocks to an extremely long OD sequence which includes all time stmaps in previous few days or weeks is computationally expensive and memory consuming. Although the replacement of RNN-based architectures by convolutional filters in ST-Conv blocks, the model training is still inefficient since most time points included in this kind of long time sequence do not make enough contributions to determine the value of the snapshot to be predicted. Only a small set of continuous timestamps in each previous day is required to capture the long-term periodicity. Assuming the predicted time index is $(d, t+j)$, we pick p_2 time intervals from $(t+j-(p_2-1)/2)$ to $(t+j+(p_2-1)/2)$ at each day $d-\varphi$ with $t+j$ in the middle for $\varphi = 1, \ldots, q$. The p_2 timestamps are used at each day $d-\varphi$ instead of a single time point $(d-\varphi, t+j)$ since the long-term periodicity is not strict and may vary in a small range around $t+j$. This slight time shifting is caused by unstable traffic peaks, holidays and extreme weather conditions among different days.

To capture the shift of the long-term periodicity, we modify the periodically shifted attention mechanism proposed by Yao et al. (2018), which is originally designed for RNN-based model, to work for the CNN-bsed ST-Conv blocks here. For each day $(d-\varphi)$, we apply $(p_2 - n^0_{LT})/(2K^0_{LT} - 2)$ ST-Conv blocks to the day-level p_2-length sequential OD flow data indexed by $\{o_{d-\varphi,t+j-(p_2+1)/2}; \ldots; o_{d-\varphi,t+j+(p_2+1)/2}\}$ to reduce the sequence length from p_2 to n^0_{LT}. We let the two TGCNNs in all the $(p_2 - n^0_{LT})/(2K^0_{LT} - 2)$ ST-Conv blocks have the same filter size $1 \times 1 \times K^0_{LT}$. The propagation rule of the n-th ST-Conv blocks is defined as:

$$z^{n+1}_{d-\varphi} = G_1(\gamma^{01}_{LT}) *_\tau [A(\theta^0_{LT}) *_L \{G_0(\gamma^{00}_{LT}) *_\tau z^n_{d-\varphi}\}] \tag{5.7}$$

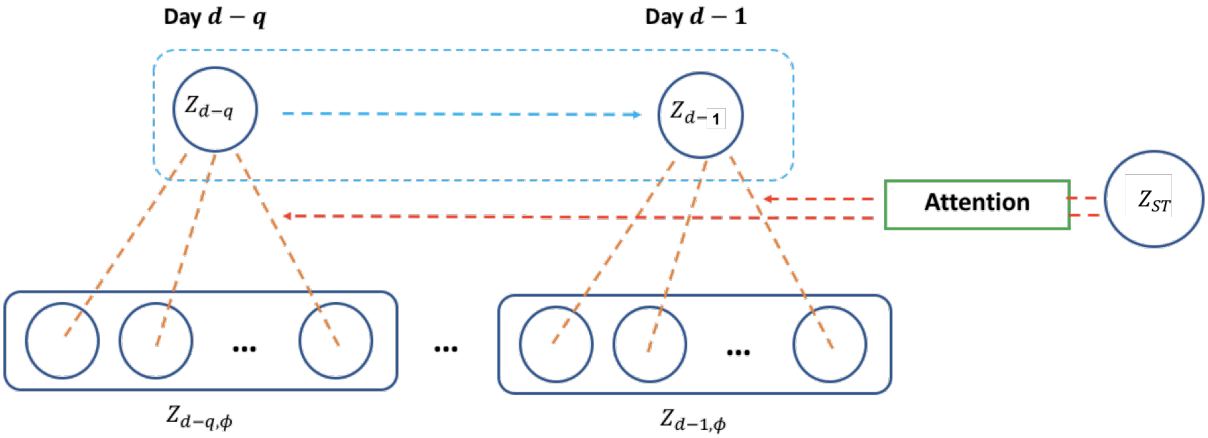

Figure 5.6: The architecture of Periodically Shifted Attention

with $z^n_{d-\varphi} \in R^{N\times N\times\{p_2-2(n-1)(K^0_{LT}-1)\}\times\tilde{c}}$ and $z^{n+1}_{d-\varphi} \in R^{N\times N\times\{p_2-2n(K^0_{LT}-1)\}\times\tilde{c}}$ being the input and output, respectively. Specifically, $z^1_{d-\varphi}$ is the original OD flow data at day $d-\varphi$ of size $N\times N\times p_2\times 1$. All SACN and TGCNN layers use $\tilde{c}$ convolutional filters and $(\theta^0_{LT}, \gamma^{00}_{LT}, \gamma^{01}_{LT})$ is the parameter set.

We denote the day-level features of day $(d-\varphi)$ captured by $(p_2 - n^0_{LT})/(2K^0_{LT}-2)$ ST-Conv blocks as $\tilde{z}_{d-\varphi} \in R^{N\times N\times n^0_{LT}\times\tilde{c}}$, where $\tilde{z}^{ij}_{d-\varphi,\phi} \in R^{\tilde{c}\times 1}$ denotes the ϕ-th element along the time axis for the OD flow from v_i to v_j. We let $z^{ij}_{ST} \in R^{c_{ST}\times 1}$ be the learned short-term representation at the OD flow from v_i to v_j. Then, a day-level output $z^{ij}_{d-\varphi}$ can be obtained by summing up all the n^0_{LT} $\tilde{z}^{ij}_{d-\varphi,\phi}$'s by the weights which measure their similarities with z^{ij}_{ST}:

$$z^{ij}_{d-\varphi} = \sum_{\phi=1}^{n^0_{LT}} \beta^{ij}_{d-\varphi,\phi}\tilde{z}^{ij}_{d-\varphi,\phi}, \tag{5.8}$$

where $\beta^{ij}_{d-\varphi,\phi}$ is the weight function of quantifying the similarity between $\tilde{z}^{ij}_{d-\varphi,\phi}$ and z^{ij}_{ST} based on a score function $\text{score}(\tilde{z}^{ij}_{d-\varphi,\phi}, z^{ij}_{ST})$, which is defined as:

$$\beta^{ij}_{d-\varphi,\phi} = \frac{\exp(\text{score}(\tilde{z}^{ij}_{d-\varphi,\phi}, z^{ij}_{ST}))}{\sum_{\phi'}\exp(\text{score}(\tilde{z}^{ij}_{d-\varphi,\phi'}, z^{ij}_{ST}))}. \tag{5.9}$$

Moreover, score($\tilde{z}^{ij}_{d-\varphi,\phi}, z^{ij}_{ST}$) is defined as

$$v_\phi^T \tanh(W_1 \tilde{z}^{ij}_{d-\varphi,\phi} + W_2 z^{ij}_{ST} + b_s), \tag{5.10}$$

where $W_1 \in R^{\tilde{c}\times\tilde{c}}, W_2 \in R^{\tilde{c}\times c_{ST}}$, and $v_\phi \in R^{\tilde{c}\times 1}$ are learned projection matrices, and b_s is the added bias term. We let $z_{d-\varphi} = (z^{ij}_{d-\varphi}) \in R^{N\times N\times\tilde{c}}$ denote the day-level output including all the N^2 OD flows.

We then concatenate the q $z_{d-\varphi}$'s along a new additional axis in the third dimension as

$$z^0_{LT} = \text{Concat}^1_{\varphi=q} z_{d-\varphi} \tag{5.11}$$

to build a new day-wise time series $z^0_{LT} \in R^{N\times N\times q\times\tilde{c}}$ of length q.

Finally, we apply another set of $(q-1)/(2K^1_{LT}-2)$ ST-Conv blocks to the day-wise sequence data generated by (5.11) to capture the long-term spatial-temporal representations. The detailed formulation for the n-th ST-Conv block is defined as

$$z^{n+1}_{LT} = G_1(\gamma^{11}_{LT}) *_\tau [A(\theta^1_{LT}) *_L \{G_0(\gamma^{10}_{LT}) *_\tau z^n_{LT}\}]. \tag{5.12}$$

The filter size is $1\times 1\times K^1_{LT}$ for all included TGCNNs. The final output of the last $(q-1)/(2K^1_{LT}-2)$-th ST-Conv block will be the learned long-term spatial-temporal representation, which is denoted by $z_{LT} \in R^{N\times N\times c_{LT}}$, where c_{LT} is the number of feature channels. The whole mechanism is illustrated in Figure 5.6.

5.3.5 Final prediction layer

We concatenate the short-term and long-term spatial-temporal representations z_{ST} and z_{LT} together along the feature axis as $X = z_{ST} \oplus z_{LT} \in R^{N\times N\times\mathcal{C}}$, where $\mathcal{C} = c_{ST} + c_{LT}$. Then, X is modified to a 2D tensor $\tilde{X} \in R^{N^2\times\mathcal{C}}$ by flattening the first two dimensions while keeping the third one. We apply a fully connected layer to the $\mathcal{C}$ feature channels together with an

element-wise non-linear sigmoid function to get the final predictions for all the N^2 OD flows:

$$\widehat{O}_{d,t+j} = \text{sigmoid}(W\tilde{X} + b), \tag{5.13}$$

where W and b are projection matrix and bias term, respectively. The 'sigmoid' activation ensures that all predictions fall into $(0, 1)$ since we normalize the original OD flow data to increase the training stability of the STOD model. The predictions will be denormalized later to get the actual value.

5.3.6 Optimization

We use L_2 loss to build the objective loss function during the training. The loss function is defined as:

$$L(\xi) = ||\widehat{o}_{d,t+j} - o_{d,t+j}||^2, \tag{5.14}$$

where ξ contains all the parameters to be learnt by using our STDO model. All the N^2 elements in both $\widehat{o}_{d,t+j}$ and $o_{d,t+j}$ here are in the range $(0, 1)$. The model is optimized via Backpropagation Through Time (BPTT) and Adam (Kingma and Ba, 2014). The whole architecture of our model is realized using Tensorflow (Abadi et al., 2016) and Keras (Chollet et al., 2015).

5.4 Experiment

In this section, we compare the proposed STOD model with some state-of-the-art approaches for traffic flow predictions. All compared methods are classified into traditional statistical methods and deep-learning based approaches. We use the order data with origin and destination information collected by a ride-sharing platform in order to examine the finite sample performance of OD flow predictions for each method.

5.4.1 Dataset Description

We employ a large-scale demand dataset obtained from a ride-sharing platform to do all the experiments. The dataset contains all customer requests received by the platform from 04/01/2018 to 06/30/2018 in a big city. The main urban area is divided into around 300

non-overlapping hexagonal sub-regions with radius being 2 km, $N = 50$ of which with the largest customer demands are selected to build the vertex set V. In total 2500 OD flows are generated based on the $|V| = 50$ sub-regions.

We split the whole dataset into two parts. The data from 04/01/2018 to 06/16/2018 is used for model training, while the other part from 06/17/2017 to 06/30/2017 (14 days) serves as the testing set. The first two and half months of OD flow data is further divided in half to the training and validation sets. The size ratio between the two sets is around 4:1. We let 30 min be the length of each timestamp and the value of the OD flow from v_i to v_j is the cumulative number of customer requests. We make predictions for all the 50^2 OD flows in the incoming 1st, 2nd, 3rd 30 minutes (i.e. $t+1, t+2, t+3$) by each compared method, given the historical data with varied (p_1, p_2) combinations. For those model settings incorporating long-term information, we trace back $q = 3$ days to capture the time periodicity.

5.4.2 Evaluation Metric

To evaluate the performance of each method, we use Rooted Mean Square Error (RMSE) defined as

$$\text{RMSE} = \sqrt{\frac{1}{N^2 * |\mathcal{T}_0|} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{(d,t) \in \mathcal{T}_0} (o_{d,t}^{ij} - \widehat{o}_{d,t}^{ij})^2}, \tag{5.15}$$

where $o_{d,t}^{ij}$ and $\widehat{o}_{d,t}^{ij}$ are the true value and prediction at the OD flow from vertex v_i to vertex v_j in the t-th timestamp of day d, respectively. The $\mathcal{T}_0$ is the set containing all the predicted time points in the testing data. Therefore, the size of the testing set is $N^2 * |\mathcal{T}_0|$.

5.4.3 Compared Methods

All state-of-the-art methods to be compared are listed as follows, some of which are modified to work for the OD flow data. We only consider latent models, that is to say no external covariates are allowed, while only the historical OD flow data is used to extract the hidden spatial-temporal features.

- **Historical average (HA)**: HA predicts the demand amount at each OD flow by the average value of the same $(t+j)$-th time index in previous 5 days.

- **Autoregressive integrated moving average (ARIMA)**: ARIMA is a class of model that captures a suite of different standard temporal structures in time series data combining moving average and autoregressive components.

- **Support Vector Machine Regression (SVMR)**: SVMR is an nonparametric approach for classification and regression relying on kernel functions.

- **Latent Space Model for Road Networks (LSM-RN)** (Deng et al., 2016): LSM-RN learns the temporal connections across time based on learned decomposition of the dynamic demand flow matrices.

- **Dense + BiLSTM** (Altché and de La Fortelle, 2017): The architecture consists of two bidirectional LSTM layers (learn from both 'past' and 'future') and two dense layers, which model temporal dependency, but capture little spatial information.

- **Spatiotemporal Recurrent Convolutional Networks (SRCN)** (Yu et al., 2017): SRCN treats the dynamic OD flow matrices as a sequence of images in the size $N \times N$. The spatial dependencies is captured by CNNs, and the temporal dynamics is learned by LSTMs

- **STOD**: Our model.

5.4.4 Experiment Setting

For the deep-learning based approaches, we normalized the original OD flow data in the training set to $(0, 1)$ using Max-Min normalization, where the upper and lower bounds are used to denormalize the predictions of testing data to get the actual values. We tune the hyperparameters of each compared model to obtain the optimal prediction performance. For fair comparison, a two-layer architecture is used by all the deep-learning based methods to extract the spatial patterns inside the OD flow data. We set the filter size of all deep learning layers in both spatial and temporal space to be 64, including the SACNs and TGCNNs in our STOD model. Each individual training batch contains 10 randomly sampled timestamps

and all the 50^2 OD flows in each snapshot. The initial liearning rate is set to be $1e-4$ with a decay rate $1e-6$. We use early stopping for all the deep learning-based methods where the training process is terminated when the RMSE over validation set has not been improved for 10 successive epochs.

5.4.5 Results

Comparison with state-of-the-art methods. In this experiment, we set the length of short-term OD flow sequence to be $p_1 = 9$ (i.e., previous 4.5 hours), $q = 3$ for long-term data which covers the three most recent days, and the length of each day-level time series $p_2 = 5$ to capture the periodicity shifting (one hour before and after the predicted time index).

Table 5.1 summarizes the finite sample performance for all the competitive methods and our STOD model in terms of the prediction RMSE on the testing data. Our model outperforms all other methods on the testing data with the lowest RMSE (2.44/2.59/2.69), achieving (6.51%/6.83%/7.24%) improvement over the second best method 'SRCN'. This demonstrates the advantages of our spatial-temporal architecture and long-term periodicity mechanism in modelling the dynamic evolution of the OD flow networks. The improvement increases as the predicting scope increases since our model captures the long-term periodicity. 'Dense + BiLSTM' outperforms traditional approaches by more precisely learning the temporal dependency using deep learning architecture, but it fails to model the underlying graph structure of OD flow data. Both 'ARIMA' and 'LSM-RN' perform poorly, even much worse than HA, indicating that they do not capture enough short-term spatial-temporal features to get the evolution trend of OD flow data.

ACN VS standard local CNN. In this experiment, we will show that our proposed SACN outperforms standard CNNs in capturing the hidden network structure of the OD flow data. Given the model setting that $N = 50$ are used to build the dynamic OD flow matrices, the number of pixels being covered by SACN at each single snapshot is $50 \times 4 = 200$. For fair comparison, the largest receptive filed of standard CNN should be no bigger than a 15×15 window, which includes 225 elements each time. Five different kernel sizes are studied,

Table 5.1: Comparison with State-of-art methods

	RMSE		
Method	30 min	60 min	90 min
HA		4.02	
ARIMA	5.64	6.01	6.49
LSVR	3.53	3.95	4.06
LSM-RN	5.73	6.36	6.74
Dense + BiLSTM	3.08	3.59	3.99
SRCN	2.61	2.78	2.90
STOD	**2.44**	**2.59**	**2.69**

which are 5×5, 8×8, 11×11, 14×14 and 15×15, respectively. We replace SCAN in our model by standard CNN to fairly compare its performance. All hyper-parameters are fixed but only the kernel size of CNNs being changed. Moreover, we only consider the baseline short-term mode of STOD model while ignoring the long-term information. As Figure 5.7 (a) illustrates, standard CNN achieves the best performance with the smallest RMSE = 2.64 on testing data when the filter size being 11×11, which is still higher than that using SACN with RMSE = 2.54. Specifically, RMSE increases when the receptive field is getting larger than 11×11 since the since the spatial correlations among the most related OD flows (sharing common origin or destination nodes) are smoothed with the increase in the filter size ($(8 \times 2 - 1)/64 > (14 \times 2 - 1)/196$). This experiment shows that treating the dynamic demand matrix as an image, and applying standard CNN filters does not capture enough spatial correlations among related OD flows without considering their topological connections from the perspective of graphs. For more details, please refer to Figure 5.7 (a).

Comparison with variants of STOD. Table 5.2 shows the finite sample performance of our proposed model STOD and its different variants. We can see that the complete model incorporating the long-term information (RMSE = 2.49) outperforms the baseline setting only using short-term data (RMSE = 2.54). This shows the necessity of modeling the seasonal temporal patterns. On the other hand, the model using the attention mechanism (RMSE = 2.44) outperforms the one without using it (RMSE = 2.49). It indicates that the periodically shifted attention can capture the shifting of the day-wise periodicity and extract

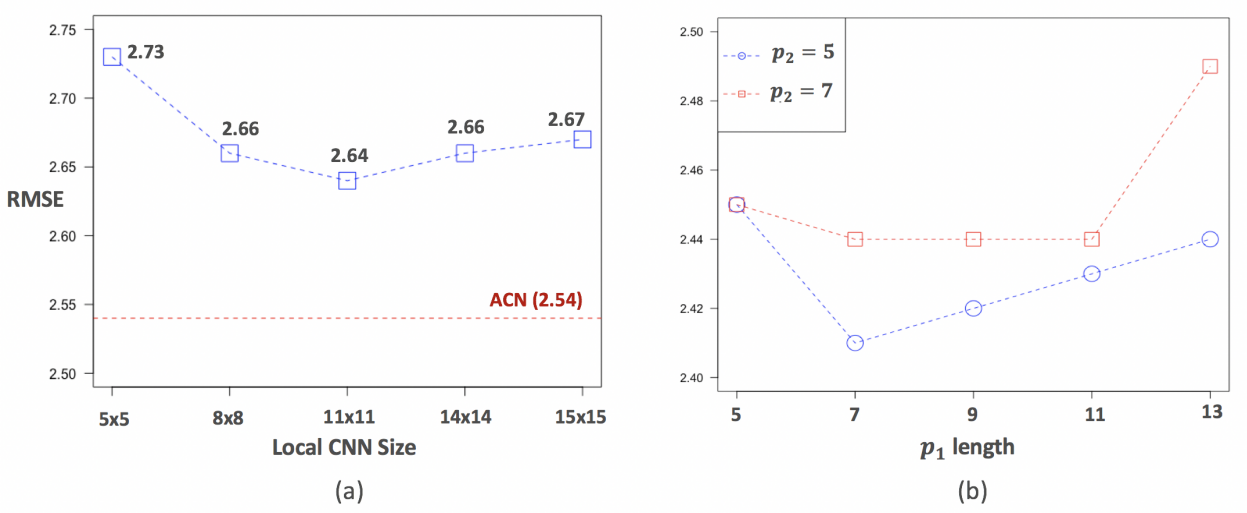

Figure 5.7: (a) RMSE on testing data with respect to ACN and standard CNN using different kernel sizes. (b) RMSE on testing data with respect to STOD with different p_1 and p_2 combinations.

Table 5.2: Evaluation of STOD and its variants

	RMSE		
Method	30 min	60 min	90 min
ACN + GCNN	2.54	2.71	2.83
ACN + GCNN + long term	2.49	2.63	2.72
ACN + GCNN + Attention	**2.44**	**2.59**	**2.69**

more seasonal patterns to improve prediction accuracy.

Figure 5.7 (b) compares RMSE on testing data by STOD model with different data settings. Varied combinations of the short-term sequence length p_1 and the long-term day-level sequence length p_2 are studied. We can see that the best performance is achieved as $(p_1, p_2) = (7, 5)$ with RMSE $= 2.41$. Specifically, settings with different p_1's under $p_2 = 5$ consistently outperform those under $p_2 = 7$. It may demonstrate that the shift can usually be captured within a short time range, while a longer time sequence may smooth the significance. Table 5.3 provides the detailed prediction results for each data setting.

5.5 Discussion

We introduces a hierarchical spatial-temporal architectures STOD for predictions of OD flow data. Compared to state-of-the-art deep learning based approaches which models the

Table 5.3: Comparison of STOD under different p_1, p_2 combinations

p_2	(K^0_{LT}, K^1_{LT})	p_1	(K^0_{ST}, K^1_{ST})	RMSE
5	$(2, 2)$	5	$(2, 2)$	2.45
		7	$(\mathbf{2}, \mathbf{3})$	**2.41**
		9	$(3, 3)$	2.42
		11	$(3, 4)$	2.43
		13	$(4, 4)$	2.43
7	$(3, 2)$	5	$(2, 2)$	2.45
		7	$(2, 3)$	2.44
		9	$(3, 3)$	2.44
		11	$(3, 4)$	2.44
		13	$(4, 4)$	2.49

OD flow matrix as an image, STOD captures the sptial features from the respective of graphs by using an irregular CNN filters. Our model jointly learns spatial-temporal representations, and captures the shift of long-term periodicity by an attention-based mechanism. We evaluate our model on a large-scale customer requests dataset in OD flow format from the ride-sharing platform, and the experimental results demonstrates that STOD outperforms many state-of-the-art methods.

APPENDIX A: APPENDIX FOR CHAPTER 2

A.1 Proofs and Explicit forms

A.1.1 Proof of (3.2)

Since it is assumed that $S = 1$ is independent of $(Y, \boldsymbol{X})$ given D, we have $P(\boldsymbol{X}, Y, S = 1|D) = P(\boldsymbol{X}, Y|D)P(S = 1|D)$. Therefore, we have

$$\begin{aligned}
E(Y|\boldsymbol{X}, D, S = 1) &= \int yp(y|\boldsymbol{X}, D, S = 1)\, dy \\
&= \int y\frac{p(y, \boldsymbol{X}, D, S = 1)}{p(\boldsymbol{X}, D, S = 1)}\, dy \\
&= \int y\frac{p(y, \boldsymbol{X}, S = 1|D)p(D)}{p(\boldsymbol{X}, S = 1|D)p(D)}\, dy \\
&= \int y\frac{p(y, \boldsymbol{X}|D)p(S = 1|D)p(D)}{p(\boldsymbol{X}|D)p(S = 1|D)p(D)}\, dy \\
&= \int y\frac{p(y, \boldsymbol{X}|D)p(D)}{p(\boldsymbol{X}|D)p(D)}\, dy = E(Y|\boldsymbol{X}, D).
\end{aligned}$$

A.1.2 Proof of (3.3)

Since $P(D, \boldsymbol{X}, S = 1) = P(\boldsymbol{X}, S = 1|D)P(D) = P(\boldsymbol{X}|D)P(S = 1, D)$, we have

$$\begin{aligned}
\frac{\Pi_j(\boldsymbol{X})}{\Pi_0(\boldsymbol{X})} \cdot \frac{\tilde{\pi}_0}{\tilde{\pi}_j} &= \frac{P(D = j|\boldsymbol{X}, S = 1)P(D = 0|S = 1)}{P(D = 0|\boldsymbol{X}, S = 1)P(D = j|S = 1)} \\
&= \frac{P(D = j, \boldsymbol{X}, S = 1)P(D = 0, S = 1)}{P(D = 0, \boldsymbol{X}, S = 1)P(D = j, S = 1)} \\
&= \frac{P(\boldsymbol{X}|D = j)}{P(\boldsymbol{X}|D = 0)} = \frac{P(\boldsymbol{X}, D = j)/P(D = j)}{P(\boldsymbol{X}, D = 0)/P(D = 0)} \\
&= \frac{P(D = j|\boldsymbol{X})P(\boldsymbol{X})}{P(D = 0|\boldsymbol{X})P(\boldsymbol{X})} \cdot \frac{P(D = 0)}{P(D = j)} \\
&= \frac{P_j(\boldsymbol{X})}{P_0(\boldsymbol{X})} \cdot \frac{\tilde{p}_0}{\tilde{p}_j},
\end{aligned}$$

for $j = 0, 1, \ldots, J - 1$.

A.1.3 Proof of (3.14)

We have

$$\partial_\theta U(\widehat{\boldsymbol{\theta}},\widehat{\boldsymbol{\varphi}}) = -\sum_{i=1}^{N} \boldsymbol{A}_{i(1)}\boldsymbol{A}_{i(1)}^T,\ \partial_{\boldsymbol{\varphi}} U(\widehat{\boldsymbol{\theta}},\widehat{\boldsymbol{\varphi}}) = -\sum_{i=1}^{N} \boldsymbol{A}_{i(1)}\boldsymbol{A}_{i(2)}^T, \text{and } \partial^2_{\boldsymbol{\varphi}} L(\boldsymbol{\varphi}) = -\sum_{i=1}^{N} \boldsymbol{A}_{i(3)},$$

where $\boldsymbol{A}_{i(1)}^T = (\boldsymbol{X}_i^T,\ \{1(D_i=1) - \widehat{P}_1(\boldsymbol{X}_i,\widehat{\boldsymbol{\varphi}})\}\boldsymbol{X}_i^T,\ \{1(D_i=2) - \widehat{P}_2(\boldsymbol{X}_i,\widehat{\boldsymbol{\varphi}})\}\boldsymbol{X}_i^T)$,

$$\boldsymbol{A}_{i(2)}^T = (\{\boldsymbol{X}_i^T\widehat{\boldsymbol{\Gamma}}_1\widehat{P}_{i1}(1-\widehat{P}_{i1}) - \boldsymbol{X}_i^T\widehat{\boldsymbol{\Gamma}}_2\widehat{P}_{i1}\widehat{P}_{i2}\}\boldsymbol{X}_i^T,\ \{-\boldsymbol{X}_i^T\widehat{\boldsymbol{\Gamma}}_1\widehat{P}_{i1}\widehat{P}_{i2} + \boldsymbol{X}_i^T\widehat{\boldsymbol{\Gamma}}_2\widehat{P}_{i2}(1-\widehat{P}_{i2})\}\boldsymbol{X}_i^T),$$

$$\text{and} \quad \boldsymbol{A}_{i(3)} = \begin{pmatrix} \widehat{\Pi}_{i1}(1-\widehat{\Pi}_{i1})\boldsymbol{X}_i\boldsymbol{X}_i^T & -\widehat{\Pi}_{i1}\widehat{\Pi}_{i2}\boldsymbol{X}_i\boldsymbol{X}_i^T \\ -\widehat{\Pi}_{i1}\widehat{\Pi}_{i2}\boldsymbol{X}_i\boldsymbol{X}_i^T & \widehat{\Pi}_{i2}(1-\widehat{\Pi}_{i2})\boldsymbol{X}_i\boldsymbol{X}_i^T \end{pmatrix}.$$

Moreover, $\widehat{\Pi}_{ij}$ and $\widehat{P}_{ij}$ denote $\widehat{\Pi}_j(\boldsymbol{X}_i;\widehat{\boldsymbol{\varphi}})$, and $\widehat{P}_j(\boldsymbol{X}_i;\widehat{\boldsymbol{\varphi}})$ for the i-th subject, respectively. Finally, we have

$$\widehat{\text{Cov}}\begin{pmatrix} \frac{1}{\sqrt{N}}U(\widehat{\boldsymbol{\theta}},\widehat{\boldsymbol{\varphi}}) \\ \frac{1}{\sqrt{N}}\partial_{\boldsymbol{\varphi}} L(\widehat{\boldsymbol{\varphi}}) \end{pmatrix} = \frac{1}{N}\sum_{i=1}^{N}\begin{pmatrix} U_i(\widehat{\boldsymbol{\theta}},\widehat{\boldsymbol{\varphi}}) - \bar{U}(\widehat{\boldsymbol{\theta}},\widehat{\boldsymbol{\varphi}}) \\ \partial_{\boldsymbol{\psi}} L_i(\widehat{\boldsymbol{\varphi}}) - \overline{\partial_{\boldsymbol{\varphi}} L(\widehat{\boldsymbol{\varphi}})} \end{pmatrix}\begin{pmatrix} U_i(\widehat{\boldsymbol{\theta}},\widehat{\boldsymbol{\varphi}}) - \bar{U}(\widehat{\boldsymbol{\theta}},\widehat{\boldsymbol{\varphi}}) \\ \partial_{\boldsymbol{\varphi}} L_i(\widehat{\boldsymbol{\varphi}}) - \overline{\partial_{\boldsymbol{\varphi}} L(\widehat{\boldsymbol{\varphi}})} \end{pmatrix}^T.$$

Moreover, we have

$$\partial_{\boldsymbol{\theta}} U(\widehat{\boldsymbol{\theta}},\widehat{\boldsymbol{\varphi}}) = -\sum_{i=1}^{N} \boldsymbol{B}_{i(1)}\boldsymbol{B}_{i(1)}^T,\ \partial_{\boldsymbol{\varphi}} U(\widehat{\boldsymbol{\theta}},\widehat{\boldsymbol{\varphi}}) = -\sum_{i=1}^{N} \boldsymbol{B}_{i(1)}\boldsymbol{B}_{i(2)}^T, \text{and } \partial^2_{\boldsymbol{\varphi}} L(\boldsymbol{\varphi}) = -\sum_{k=1}^{2}\sum_{i=1}^{N} \boldsymbol{B}_{i(3)}^{(k)} 1(m_i = k)$$

where $\boldsymbol{B}_{i(1)}^T = (\boldsymbol{X}_i^T,\ \{1(D_i=1) - \widehat{P}_1(\boldsymbol{X}_i,\widehat{\boldsymbol{\varphi}})\}\boldsymbol{X}_i^T,\ \{1(D_i=2) - \widehat{P}_2(\boldsymbol{X}_i,\widehat{\boldsymbol{\varphi}})\}\boldsymbol{X}_i^T)$,

$$\boldsymbol{B}_{i(2)}^T = (\{\boldsymbol{X}_i^T\widehat{\boldsymbol{\Gamma}}_1\widehat{P}_{i1}(1-\widehat{P}_{i1}) - \boldsymbol{X}_i^T\widehat{\boldsymbol{\Gamma}}_2\widehat{P}_{i1}\widehat{P}_{i2}\}\boldsymbol{X}_i^T,\ \{-\boldsymbol{X}_i^T\widehat{\boldsymbol{\Gamma}}_1\widehat{P}_{i1}\widehat{P}_{i2} + \boldsymbol{X}_i^T\widehat{\boldsymbol{\Gamma}}_2\widehat{P}_{i2}(1-\widehat{P}_{i2})\}\boldsymbol{X}_i^T),$$

$$\boldsymbol{B}_{i(3)}^{(m)} = \begin{pmatrix} \widehat{\Pi}_{i1}^{(m)}(1-\widehat{\Pi}_{i1}^{(m)})\boldsymbol{X}_i\boldsymbol{X}_i^T & -\widehat{\Pi}_{i1}^{(m)}\widehat{\Pi}_{i2}^{(m)}\boldsymbol{X}_i\boldsymbol{X}_i^T \\ -\widehat{\Pi}_{i1}^{(m)}\widehat{\Pi}_{i2}^{(m)}\boldsymbol{X}_i\boldsymbol{X}_i^T & \widehat{\Pi}_{i2}^{(m)}(1-\widehat{\Pi}_{i2}^{(m)})\boldsymbol{X}_i\boldsymbol{X}_i^T \end{pmatrix}$$

and

$$\widehat{\mathrm{Cov}}\begin{pmatrix} \frac{1}{\sqrt{N}}U(\widehat{\boldsymbol{\theta}},\widehat{\boldsymbol{\varphi}}) \\ \frac{1}{\sqrt{N}}\partial_{\boldsymbol{\varphi}}L(\widehat{\boldsymbol{\varphi}}) \end{pmatrix} = \frac{1}{N}\sum_{i=1}^{N}\begin{pmatrix} U_i(\widehat{\boldsymbol{\theta}},\widehat{\boldsymbol{\varphi}}) - \bar{U}(\widehat{\boldsymbol{\theta}},\widehat{\boldsymbol{\varphi}}) \\ \partial_{\boldsymbol{\varphi}}L_i(\widehat{\boldsymbol{\varphi}}) - \overline{\partial_{\boldsymbol{\varphi}}L(\widehat{\boldsymbol{\varphi}})} \end{pmatrix}\begin{pmatrix} U_i(\widehat{\boldsymbol{\theta}},\widehat{\boldsymbol{\varphi}}) - \bar{U}(\widehat{\boldsymbol{\theta}},\widehat{\boldsymbol{\varphi}}) \\ \partial_{\boldsymbol{\varphi}}L_i(\widehat{\boldsymbol{\varphi}}) - \overline{\partial_{\boldsymbol{\varphi}}L(\widehat{\boldsymbol{\varphi}})} \end{pmatrix}^T.$$

A.2 D with more than three categories

In this part, we extend the case of three groups to the more general case of J groups. Following the proof of (2), we still have that $\tilde{\mu}(\boldsymbol{X},i) = E(Y|\boldsymbol{X},D=i) = E(Y|\boldsymbol{X},D=i,S=1)$ holds for $i=0,1,\ldots,J-1$. Then, we have the relation between $\mu(\boldsymbol{X}) = E(Y|\boldsymbol{X})$ and $\tilde{\mu}(\boldsymbol{X},i) = E(Y|\boldsymbol{X},D=i)$ as

$$\mu(\boldsymbol{X}) = \sum_{i=0}^{J-1}\tilde{\mu}(\boldsymbol{X},i) * P(D=i|\boldsymbol{X}) \tag{A.1}$$

and with $\sum_{i=0}^{J-1}P(D=i|\boldsymbol{X}) = 1$, we still have

$$\tilde{\mu}(\boldsymbol{X},i) = \mu(\boldsymbol{X}) + \sum_{j\neq i}P(D=j|\boldsymbol{X})(\tilde{\mu}(\boldsymbol{X},i) - \tilde{\mu}(\boldsymbol{X},j)) \tag{A.2}$$

When $i>0$, by assuming $\gamma_i(\boldsymbol{X}) = \tilde{\mu}(\boldsymbol{X},i) - \tilde{\mu}(\boldsymbol{X},0)$, (A.1) can be rewritten as

$$\begin{aligned}
\tilde{\mu}(\boldsymbol{X},i) &= \mu(\boldsymbol{X}) + \sum_{j\neq i}P(D=j|\boldsymbol{X})\{\tilde{\mu}(\boldsymbol{X},i) - \tilde{\mu}(\boldsymbol{X},j)\} \\
&= \mu(\boldsymbol{X}) + P(D=0|\boldsymbol{X})\{\tilde{\mu}(\boldsymbol{X},i) - \tilde{\mu}(\boldsymbol{X},0)\} + \sum_{j\neq i,0}P(D=j|\boldsymbol{X})\{\tilde{\mu}(\boldsymbol{X},i) - \tilde{\mu}(\boldsymbol{X},j)\} \\
&= \mu(\boldsymbol{X}) + \{1 - \sum_{k\neq 0}P(D=k|\boldsymbol{X})\}\{\tilde{\mu}(\boldsymbol{X},i) - \tilde{\mu}(\boldsymbol{X},0)\} + \sum_{j\neq i,0}P(D=j|\boldsymbol{X})\{\tilde{\mu}(\boldsymbol{X},i) - \tilde{\mu}(\boldsymbol{X},j)\} \\
&= \mu(\boldsymbol{X}) + 1 - P(D=0|\boldsymbol{X})\{\tilde{\mu}(\boldsymbol{X},i) - \tilde{\mu}(\boldsymbol{X},0)\} - \sum_{j\neq i,0}P(D=j|\boldsymbol{X})\{\tilde{\mu}(\boldsymbol{X},j) - \tilde{\mu}(\boldsymbol{X},0)\} \\
&= \mu(\boldsymbol{X}) + \sum_{k=1}^{J-1}D_k - \sum_{j\neq 0}P(D=j|\boldsymbol{X})(\tilde{\mu}\{\boldsymbol{X},j) - \tilde{\mu}(\boldsymbol{X},0)\} \\
&= \mu(\boldsymbol{X}) + \sum_{j\neq 0}\{1(i=j) - P(D=j|\boldsymbol{X})\}\{\tilde{\mu}(\boldsymbol{X},j) - \tilde{\mu}(\boldsymbol{X},0)\} \\
&= \mu(\boldsymbol{X}) + \sum_{j\neq 0}\{1(i=j) - P(D=j|\boldsymbol{X})\}\gamma_j(\boldsymbol{X})
\end{aligned}$$

When $i = 0$, since $\sum_{k=1}^{J-1} D_k = 0$, (A.1) is equivalent to

$$\begin{aligned}
\tilde{\mu}(\boldsymbol{X}, i) &= \mu(\boldsymbol{X}) + \sum_{j \neq 0} P(D = j|\boldsymbol{X})\{\tilde{\mu}(\boldsymbol{X}, 0) - \tilde{\mu}(\boldsymbol{X}, j)\} \\
&= \mu(\boldsymbol{X}) + \sum_{j \neq 0} \{1(i = j) - P(D = j|\boldsymbol{X})\}\{\tilde{\mu}(\boldsymbol{X}, j) - \tilde{\mu}(\boldsymbol{X}, 0)\} \\
&= \mu(\boldsymbol{X}) + \sum_{j \neq 0} \{1(i = j) - P(D = j|\boldsymbol{X})\}\gamma_j(\boldsymbol{X})
\end{aligned}$$

Thus, the target model becomes

$$\tilde{\mu}(\boldsymbol{X}, i) = \mu(\boldsymbol{X}) + \sum_{j \neq 0} (1(i = j) - P(D = j|\boldsymbol{X}))\gamma_j(\boldsymbol{X}). \tag{A.3}$$

A.3 Simulations with multiple SNPs

The simulation datasets with multiple SNPs were generated according to steps given below. Moreover, we also consider two settings as the two-SNP case:

A.3.1 Setting One

(i) Generate a non-genetic covariate $C \sim N(0, 1)$ for each subject.

(ii) Generate $N_g = 500$ SNP-level genetic variables $\boldsymbol{G} = \{G_1, G_2, \ldots, G_{500}\}$ with MAF for each G_i sampled according to uniform distribution $U(0.2, 0.3)$. Then we randomly select 10 SNPs from set $\boldsymbol{G}$ as causal SNPs, denoted as $\boldsymbol{G}_c$.

(iii) Generate the primary trait D according to the following multinomial logistic model:

$$\log\left(\frac{P(D = j|\boldsymbol{X})}{P(D = 0|\boldsymbol{X})}\right) = \boldsymbol{X}^T\boldsymbol{\varphi}_j \text{ for } j = 1, 2,$$

where $\boldsymbol{X}^T = (1, C, \boldsymbol{G}_c)$, and we choose $\boldsymbol{\varphi}_1 = \boldsymbol{\varphi}_2$ to make the global prevalence of groups 0, 1, and 2 be 10%, 15%, and 75%, respectively.

(iv) Generate the secondary phenotype Y for each subject according to (A.4) as follows:

$$Y = \beta_0 + \boldsymbol{\beta}_1^T \boldsymbol{X} + \sum_{j=1}^{2} \{1(D = j) - P_j(\boldsymbol{X})\} \boldsymbol{\gamma}_j^T \boldsymbol{X} + \epsilon, \tag{A.4}$$

where each element of $\boldsymbol{\gamma}_j$ is randomly sampled from $(-0.5, 0.5)$ and $\epsilon \sim N(0, 1)$.

(v) Repeat steps (i)-(iv) to generate $(Y, \boldsymbol{X}, D)$ until we obtain a total of $N = 500,000$ observations as the whole population. Then, we randomly select 500, 1000, and 500 subjects from the $D = 0$, $D = 1$, and $D = 2$ groups, respectively, in order to build a non-random three-group sample.

A.3.2 Setting Two

(i) Generate C, $\boldsymbol{G}$, and $\boldsymbol{G}_c$ as setting one.

(ii) Generate the secondary phenotype Y for each subject according to

$$Y = \beta_0 + \boldsymbol{\beta}_1^T \boldsymbol{X} + \epsilon, \tag{A.5}$$

where $\boldsymbol{X}^T = (1, C, \boldsymbol{G}_c)$. Moreover, we set each component of $\boldsymbol{\beta}$ corresponding to each $G_i \in \boldsymbol{G}_c$ to be 0.5.

(iii) Simulate the primary trait D by using a multinomial model given by

$$\log\left(\frac{P(D = j|C, Y, \boldsymbol{G}_c)}{P(D = 0|C, Y, \boldsymbol{G}_c)}\right) = (C, Y, \boldsymbol{G}_c)\tilde{\boldsymbol{\varphi}}_j \text{ for } j = 1, 2,$$

We set $\tilde{\boldsymbol{\varphi}}_1 = \tilde{\boldsymbol{\varphi}}_2$ so that the global prevalence of groups 0, 1, and 2 are, respectively, given by 15%, 15% and 70%.

(iv) Repeat steps 1-3 until the sample size reaches 500, 000 and then sample $500(D = 0)$, $1000(D = 1)$ and $500(D = 2)$ observations from the above large pool of subjects.

A.4 The Alzheimer's Disease Neuroimaging Initiative Data

A.4.1 Sample

We used imaging and genetic data from the ADNI database obtained from phases ADNI1, ADNI2, and ADNIGO. The earliest phase, ADNI1, recruited more than 800 subjects and the latter two phases, ADNIGO and ADNI2, recruited more than 900 new subjects, and added a new cohort category, called significant memory concern (SMC). Therefore, ADNI participants represent four main groups: people with normal cognition (NC), people with early or late MCI (EMCI or LMCI), people with AD, and people with SMC.

The total number of subjects with baseline demographic information from ADNI1, ADNI2 and ADNIGO is 1737, consisting of 342 ADs, 417 NCs, 310 EMCIs, 562 LMCIs, and 106 SMCs. In ADNI1, we only include the 712 Caucasians from all 818 subjects with genetic data, among which there are 198 NCs, 352 MCIs, and 162 ADs. Moreover, we used 550 Caucasians in ADNI2 and ADNIGO, among which there are 82 ADs, 114 NCs, 201 EMCIs, 100 LMCIs, and 53 SMCs. To match the group information of ADNI1, we dropped the 53 SMC subjects and combined the EMCI and LMCI groups, leading to a three-group study. 325 subjects with genetic data finally go to the sample data, including 101 NCs, 201 MCIs and 23 ADs.

A.4.2 MRI Acquisition and Image Preprocessing

All participants enrolled in ADNI1 underwent brain scanning using a variety of 1.5 Tesla MRI scanners; whereas all participants newly enrolled in ADNIGO and ADNI2 were scanned using 3T MRI scanners. The parameters of a typical MRI protocol for ADNI1 are as follows: repetition time (TR) = 2400 ms, inversion time (TI) = 1000 ms, flip angle = 8^o, field of view (FOV) = 24 cm with a $256 \times 256 \times 170$ acquisition matrix in the $x-$, $y-$, and $z-$dimensions yielding a voxel size of $1.25 \times 1.26 \times 1.2$ mm^3 (Jack Jr et al. (2008)). The parameters of a typical MRI protocol for ADNI2 and ADNIGo are as follows: 8-channel coil, TR = 400 ms, TE = min full, flip-angle = 11^o, slice thickness = 1.2 mm, resolution = 256×256mm and FOV = 26 cm. All original and bias-corrected image files are available to the general scientific community at http://adni.loni.usc.edu/. Based on the bias-corrected T1-weighted

MRI images, we first interpolated the voxel size to $1 \times 1 \times 1mm^3$ and then used the local label learning (LLL) (Hao et al. (2014)) approach to carry out left and right hippocampal segmentation for each subject. Hao et al. (2014) showed that the LLL method leads to better segmentation results compared with most state-of-the-art label fusion methods.

A.4.3 Genotype Data

The genetic data of ADNI1 was acquired using the Human610-Quad BeadChip, while the subjects from ADNI-2 were genotyped using the Illumina Human OmniExpress BeadChip (Illumina, Inc., San Diego, CA). The original data of ADNI1 contains 620,901 genetic markers, including multiple types of genetic variants; whereas ADNI2 has 730,525 genetic markers. We then performed the following quality control procedures, including (i) call rate check per subject, (ii) gender check, (iii) sibling pair identification, and (iv) population stratification. Furthermore, SNPs were excluded from the imaging genetic analysis if they could not meet any of the following criteria: (i) call rate per SNP$\geq$ 95%, (ii) MAF $\geq$ 5%, and (iii) Hardy-Weinberg equilibrium test of $p \geq 10^{-6}$. We applied MACH-Admix software (http://www.unc.edu/ yunmli/MaCH-Admix/) (Liu et al. (2013)) to perform genotype imputation, using 1000G phase I integrated release version 3 haplotypes (http://www.1000genomes.org) (1000 Genomes Project Consortium, 2012) as a reference panel. After imputation, we obtained 7,986,566 bi-allelic markers (including SNPs and indels) in ADNI1 and 8,218,182 markers in ADNI2. Finally, we excluded those with low imputation accuracy (based on imputation output R^2), with MAF smaller than 0.05, or a p-value smaller than 10^{-6} in the Hardy-Weinberg equilibrium test, leading to 6,017,259 SNP-based markers in the final data analysis (Zhu et al., 2017).

A.5 The Boxplots of the log volumes of the left and right hippocampi in ADNI1 and ADNI2, ADNI GO

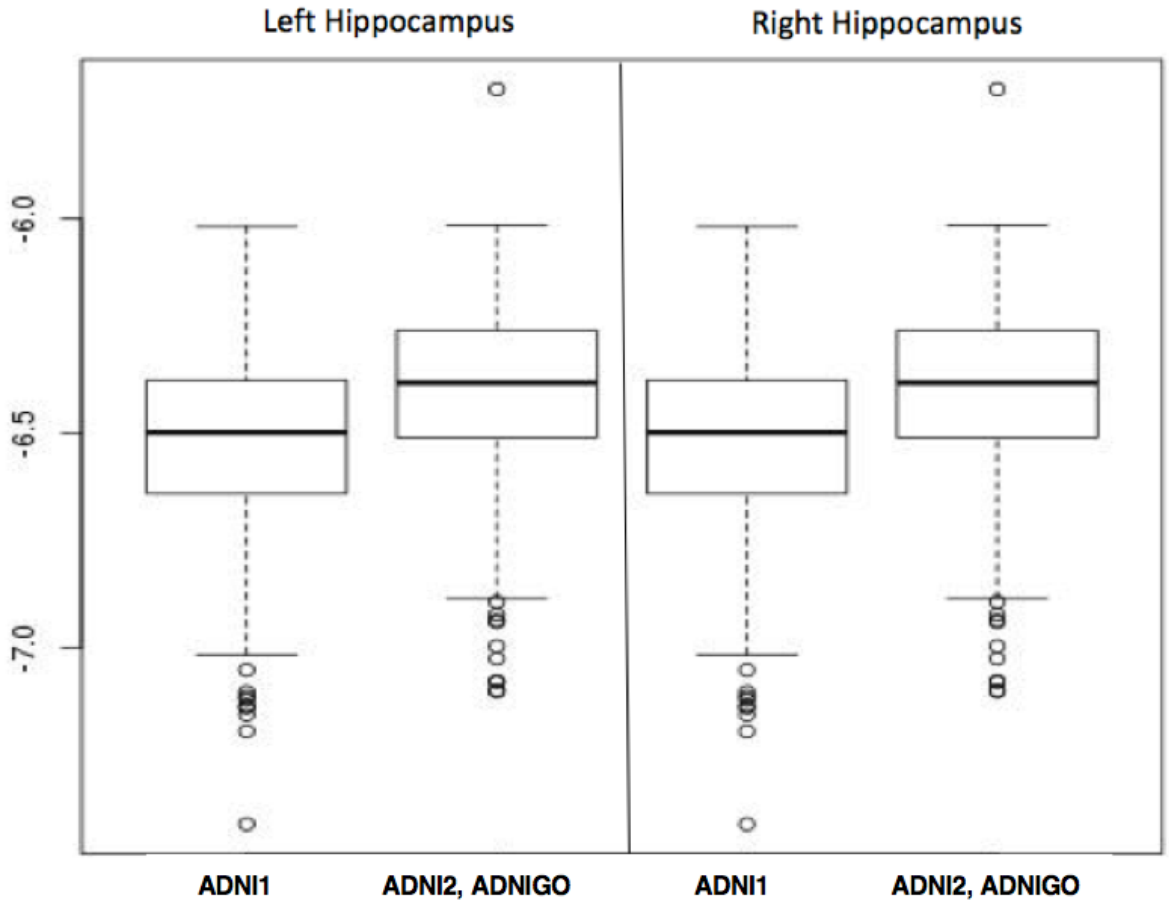

Figure A.1: The Boxplots of the log volumes of the left and right hippocampi in ADNI1 and ADNI2, ADNI GO

APPENDIX B: APPENDIX FOR CHAPTER 3

B.1 Theorem Proofs

B.1.1 Lemma and proof

The proof of Theorem 4.1 is based on the following lemma. Let $supp(\cdot)$ be the support of a domain space.

Lemma B.1. *Under the model (2), (3), or (2), (5) (main text), suppose that there exists an instrumental variable u_i in each $x_i = (z_i^T, u_i^T)^T$ such that $f(y_i|\mathscr{G}^A(\boldsymbol{x})_i)$ depends on u_i, whereas $P(r_i = 1|y_i, h(x_i))$ does not depend on u_i. We let $\boldsymbol{x} = [\boldsymbol{z}, \boldsymbol{u}]$. Our GNM model is identifiable on the PEQ space under the following sufficient Conditions (C1)-(C3):*
(C1) there exists a set $S \subset supp(Y, \boldsymbol{z})$, such that $P(r_i = 1|y_i, h(z_i); \theta_r) \neq 0$ for each i and all $(Y, \boldsymbol{z}) \in S$ and $\theta_r \in \mathcal{D}(\theta_r)$.
(C2) Denote $\theta_{r1} = (\alpha_{r1}, \gamma_1, \phi_1, \theta_{h1})^T$ and $\theta_{r2} = (\alpha_{r2}, \gamma_2, \phi_2, \theta_{h2})^T$. $P(r_i = 1|y_i, h(z_i); \theta_{r1}) = P(r_i = 1|y_i, h(z_i); \theta_{r2})$ for each i and all $(Y, \boldsymbol{z}) \in S \iff \gamma_1^T h(z_i; \theta_{h1}) = \gamma_2^T h(z_i; \theta_{h2})$ holds for all $\boldsymbol{z}$ and each z_i.
(C3) Denote $\theta_{y1} = (\alpha_1, \beta_1, \theta_{g1})^T$ and $\theta_{y2} = (\alpha_2, \beta_2, \theta_{g2})^T$. We let $\boldsymbol{x}_1 = [\boldsymbol{z}, \boldsymbol{u}_1]$ and $\boldsymbol{x}_2 = [\boldsymbol{z}, \boldsymbol{u}_2]$. If $f(y_i|\mathscr{G}^A(\boldsymbol{x}_1)_i; \theta_{y1}) f(y_i|\mathscr{G}^A(\boldsymbol{x}_2)_i; \theta_{y2}) = f(y_i|\mathscr{G}^A(\boldsymbol{x}_1)_i; \theta_{y2}) f(y_i|\mathscr{G}^A(\boldsymbol{x}_2)_i; \theta_{y1})$ holds for each i and all $(\boldsymbol{u}_1, \boldsymbol{u}_2)$ and $(Y, \boldsymbol{z}) \in S$, then $\mathscr{G}^A(\boldsymbol{x}; \theta_{g1})\beta_1 = \mathscr{G}^A(\boldsymbol{x}; \theta_{g2})\beta_2$ holds.

Proof: Suppose that the following two equations hold for all $(Y, \mathbf{z}) \in S$ and $(\mathbf{u}_1, \mathbf{u}_2)$: $\mathbf{u}_1 \neq \mathbf{u}_2$, then for each i we have

$$P(r_i = 1|y_i, h(z_i); \theta_{r1}) f(y_i|\mathscr{G}^A(\mathbf{x}_1)_i; \theta_{y1}) = P(r_i = 1|y_i, h(z_i); \theta_{r2}) f(y_i|\mathscr{G}^A(\mathbf{x}_1)_i; \theta_{y2})$$

$$P(r_i = 1|y_i, h(z_i); \theta_{r2}) f(y_i|\mathscr{G}^A(\mathbf{x}_2)_i; \theta_{y2}) = P(r_i = 1|y_i, h(z_i); \theta_{r1}) f(y_i|\mathscr{G}^A(\mathbf{x}_2)_i; \theta_{y1}) \quad \text{(B.1)}$$

Multiplying the two equations gives

$$P(r_i = 1|y_i, h(z_i); \theta_{r1}) f(y_i|\mathscr{G}^A(\mathbf{x}_1)_i; \theta_{y1}) P(r_i = 1|y_i, h(z_i); \theta_{r2}) f(y_i|\mathscr{G}^A(\mathbf{x}_2)_i; \theta_{y2})$$

$$= P(r_i = 1|y_i, h(z_i); \theta_{r2}) f(y_i|\mathscr{G}^A(\mathbf{x}_1)_i; \theta_{y2}) P(r_i = 1|y_i, h(z_i); \theta_{r1}) f(y_i|\mathscr{G}^A(\mathbf{x}_2)_i; \theta_{y1})$$

Together with condition (C1), it follows that

$$f(y_i|\mathscr{G}^A(\mathbf{x}_1)_i; \theta_{y1}) f(y_i|\mathscr{G}^A(\mathbf{x}_2)_i; \theta_{y2}) = f(y_i|\mathscr{G}^A(\mathbf{x}_1)_i; \theta_{y2}) f(y_i|\mathscr{G}^A(\mathbf{x}_2)_i; \theta_{y1})$$

holds for each i and all $(Y, \mathbf{z}) \in S$. Then from condition (C3), we have $\mathscr{G}^A(\mathbf{x}; \theta_{g1})\beta_1 = \mathscr{G}^A(\mathbf{x}; \theta_{g2})\beta_2$ for all $\mathbf{x}$, which implies $f(y_i|\mathscr{G}^A(\mathbf{x}_1)_i; \theta_{y1}) = f(y_i|\mathscr{G}^A(\mathbf{x}_1)_i; \theta_{y2})$ from (3) (main text). Then, we obtain from (B.1) that

$$P(r_i = 1|y_i, h(z_i); \theta_{r1}) = P(r_i = 1|y_i, h(z_i); \theta_{r2})$$

for each i and all $(Y, \mathbf{z}) \in S$. Together with condition (C2), we have $\gamma_1^T h(x_i; \theta_{h1}) = \gamma_2^T h(x_i; \theta_{h2})$ holds for all $\mathbf{z}$ and each z_i. and the identifiability on the PEQ space is obtained.

B.1.2 Proof of Theorem 4.1

Part (i):

Under the model (2) and (5) (main text), we prove the identifiability for the binary case when $y \in \{1, -1\}$, while all the derivations can be extended to the more general case. We need to show that for each i and all $(y_i, \mathbf{x}) \in S$,

$$\frac{1}{1 + \exp\{-\alpha_{r1} - \gamma_1^T h(x_i; \theta_{h1}) - \phi_1 y_i\}} \frac{1}{1 + \exp\{-y_i(\alpha_1 + \beta_1^T \mathscr{G}^A(\mathbf{x}; \theta_{g1})_i)\}}$$

$$= \frac{1}{1 + \exp\{-\alpha_{r2} - \gamma_2^T h(x_i; \theta_{h2}) - \phi_2 y_i\}} \frac{1}{1 + \exp\{-y_i(\alpha_2 + \beta_2^T \mathscr{G}^A(\mathbf{x}; \theta_{g2})_i)\}} \tag{B.2}$$

is equivalent to

$$\alpha_{r1} = \alpha_{r2}, \gamma_1 = \gamma_2, \phi_1 = \phi_2, \alpha_1 = \alpha_2, \beta_1 = \beta_2, \theta_{h1} = \theta_{h2}, \theta_{g1} = \theta_{g2}$$

(B.2) can be rewritten as

$$e^{-\{\alpha_{r1}+\gamma_1^T h(x_i;\theta_{h1})+\phi_1 y_i\}} + e^{-y_i\{\alpha_1+\beta_1^T \mathscr{G}^A(\mathbf{x};\theta_{g1})_i\}} + e^{-(\alpha_1 y_i+\alpha_{r1})-\phi_1 y_i-\gamma_1^T h(x_i;\theta_{h1})-\beta_1^T \mathscr{G}^A(\mathbf{x};\theta_{g1})_i y_i}$$

$$= e^{-\{\alpha_{r2}+\gamma_2^T h(x_i;\theta_{h2})+\phi_2 y_i\}} + e^{-y_i\{\alpha_2+\beta_2^T \mathscr{G}^A(\mathbf{x};\theta_{g2})_i\}} + e^{-(\alpha_2 y_i+\alpha_{r2})-\phi_2 y_i-\gamma_2^T h(x_i;\theta_{h2})-\beta_2^T \mathscr{G}^A(\mathbf{x};\theta_{g2})_i y_i} \tag{B.3}$$

Since (B.3) holds for all $(y_i, \mathbf{x})$, and from Condition (A1), the only possible solution to (B.3) is

$$\begin{cases} e^{-\{\alpha_{r1}+\gamma_1^T h(x_i;\theta_{h1})+\phi_1 y_i\}} = e^{-\{\alpha_{r2}+\gamma_2^T h(x_i;\theta_{h2})+\phi_2 y_i\}}, \\ e^{-\{\alpha_1+\beta_1^T \mathscr{G}^A(\mathbf{x};\theta_{g1})_i\}} = e^{-\{\alpha_2+\beta_2^T \mathscr{G}^A(\mathbf{x};\theta_{g2})_i\}}, \\ e^{-(\alpha_1+\alpha_{r1})-\phi_1 y_i-\gamma_1^T h(x_i;\theta_{h1})-\beta_1^T \mathscr{G}^A(\mathbf{x};\theta_{g1})_i} = e^{-(\alpha_2+\alpha_{r2})-\phi_2 y_i-\gamma_2^T h(x_i;\theta_{h2})-\beta_2^T \mathscr{G}^A(\mathbf{x};\theta_{g2})_i} \end{cases}$$

which requires

$$\alpha_{r1} = \alpha_{r2}; \phi_1 = \phi_2; \alpha_1 = \alpha_2; \beta_1^T \mathscr{G}^A(\mathbf{x};\theta_{g1})_i = \beta_2^T \mathscr{G}^A(\mathbf{x};\theta_{g2})_i; \gamma_1^T h(x_i;\theta_{h1}) = \gamma_2^T h(x_i;\theta_{h2}),$$

which concludes the identifiability on the PEQ space.

Part (ii):

Under the model (2) and (3) (main text), we prove the identifiability of the parameter when the responses y are continuous. By using Lemma (B.1), Condition (C1) holds due to (2) (main text). Condition (C2) holds due to Condition (A3) in Theorem 3.1. We next give the proof of Condition (C3). We here give the proof of $q = 1$ which can be extended to the general case.

If $f(y_i|\mathscr{G}^A(\mathbf{x}_1)_i;\theta_{y1})f(y_i|\mathscr{G}^A(\mathbf{x}_2)_i;\theta_{y2}) = f(y_i|\mathscr{G}^A(\mathbf{x}_1)_i;\theta_{y2})f(y_i|\mathscr{G}^A(\mathbf{x}_2)_i;\theta_{y1})$ holds for each

i and all $(\mathbf{u}_1, \mathbf{u}_2)$ and $(y, \mathbf{z}) \in S$, from (3) (main text), the following equation holds

$$\begin{aligned} & \beta_1^2[(\mathscr{G}^A([\mathbf{z}, \mathbf{u}_1]; \theta_{g1}))^2 - (\mathscr{G}^A([\mathbf{z}, \mathbf{u}_2]; \theta_{g1}))^2] \\ - \quad & 2(y - \alpha_1)\beta_1[\mathscr{G}^A([\mathbf{z}, \mathbf{u}_1]; \theta_{g1}) - \mathscr{G}^A([\mathbf{z}, \mathbf{u}_2]; \theta_{g1})] \\ = \quad & \beta_2^2[(\mathscr{G}^A([\mathbf{z}, \mathbf{u}_1]; \theta_{g2}))^2 - (\mathscr{G}^A([\mathbf{z}, \mathbf{u}_2]; \theta_{g2}))^2] \\ - \quad & 2(y - \alpha_2)\beta_2[\mathscr{G}^A([\mathbf{z}, \mathbf{u}_1]; \theta_{g2}) - \mathscr{G}^A([\mathbf{z}, \mathbf{u}_2]; \theta_{g2})] \end{aligned}$$

for all y. Together with Condition (A2), we have

$$\beta_1[\mathscr{G}^A([\mathbf{z}, \mathbf{u}_1]; \theta_{g1}) - \mathscr{G}^A([\mathbf{z}, \mathbf{u}_2]; \theta_{g1})] = \beta_2[\mathscr{G}^A([\mathbf{z}, \mathbf{u}_1]; \theta_{g2}) - \mathscr{G}^A([\mathbf{z}, \mathbf{u}_2]; \theta_{g2})]$$

and

$$\beta_1[\mathscr{G}^A([\mathbf{z}, \mathbf{u}_1]; \theta_{g1}) + \mathscr{G}^A([\mathbf{z}, \mathbf{u}_2]; \theta_{g1})] = \beta_2[\mathscr{G}^A([\mathbf{z}, \mathbf{u}_1]; \theta_{g2}) + \mathscr{G}^A([\mathbf{z}, \mathbf{u}_2]; \theta_{g2})].$$

It follows that

$$\beta_1 \mathscr{G}^A([\mathbf{z}, \mathbf{u}_1]; \theta_{g1}) = \beta_2 \mathscr{G}^A([\mathbf{z}, \mathbf{u}_1]; \theta_{g2})$$

and Condition (C3) holds, which concludes the proof.

B.1.3 Proof of Theorem 4.2

To prove the theorem, we use the law of iterated conditional expectation as follows:

$$\begin{aligned} E_{\theta_y}\{\sum_i \frac{r_i}{\pi(y_i, h(x_i))} l(y_i, \mathscr{G}^A(\mathbf{x})_i)\} &= E_{\theta_y}[E\{\sum_i \frac{r_i}{\pi(y_i, h(x_i))} l(y_i, \mathscr{G}^A(\mathbf{x})_i) | Y, \mathbf{x}\}] \\ &= E_{\theta_y}\{\sum_i \frac{E(r_i|Y, \mathbf{x})}{\pi(y_i, h(x_i))} l(y_i, \mathscr{G}^A(\mathbf{x})_i)\} \\ &= E_{\theta_y}\{\frac{\pi(y_i, h(x_i))}{\pi(y_i, h(x_i))} l(y_i, \mathscr{G}^A(\mathbf{x})_i)\} \\ &= E_{\theta_y}\{l(y_i, \mathscr{G}^A(\mathbf{x})_i)\} \end{aligned} \tag{B.4}$$

where (B.4) holds because

$$E(r_i|Y, \mathbf{x}) = E(r_i|y_i, x_i) = E(r_i|y_i, h(x_i)) = P(r_i = 1|y_i, h(x_i)) = \pi(y_i, h(x_i))$$

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283.

Agrawal, R., Faloutsos, C., and Swami, A. (1993). Efficient similarity search in sequence databases. In *International conference on foundations of data organization and algorithms*, pages 69–84. Springer.

Altché, F. and de La Fortelle, A. (2017). An lstm network for highway trajectory prediction. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 353–359. IEEE.

Ambrosio, L., Gigli, N., and Savaré, G. (2008). *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media.

Arpawong, T. E., Pendleton, N., Mekli, K., McArdle, J. J., Gatz, M., Armoskus, C., Knowles, J. A., and Prescott, C. A. (2017). Genetic variants specific to aging-related verbal memory: Insights from gwass in a population-based cohort. *PloS one* **12,** e0182448.

Atluri, G., Karpatne, A., and Kumar, V. (2018). Spatio-temporal data mining: A survey of problems and methods. *ACM Computing Surveys (CSUR)* **51,** 83.

Baker, S. G. and Laird, N. M. (1988). Regression analysis for categorical variables with outcome subject to nonignorable nonresponse. *Journal of the American Statistical Association* **83,** 62–69.

Bang, H. and Robins, J. M. (2005). Doubly robust estimation in missing data and causal inference models. *Biometrics* **61,** 962–973.

Beaumont, J.-F. (2000). An estimation method for nonignorable nonresponse. *Survey Methodology* **26,** 131–136.

Belkin, M., Niyogi, P., and Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research* **7,** 2399–2434.

Bickel, P. J., Klaassen, C. A., Bickel, P. J., Ritov, Y., Klaassen, J., Wellner, J. A., and Ritov, Y. (1993). *Efficient and adaptive estimation for semiparametric models*. Johns Hopkins University Press Baltimore.

Breslow, N. and Cain, K. (1988). Logistic regression for two-stage case-control data. *Biometrika* **75,** 11–20.

Breslow, N. and Powers, W. (1978). Are there two logistic regressions for retrospective studies? *Biometrics* pages 100–105.

Breslow, N. E., Robins, J. M., Wellner, J. A., et al. (2000). On the semi-parametric efficiency of logistic regression under case-control sampling. *Bernoulli* **6,** 447–455.

Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. (2013). Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203* .

Camerra, A., Palpanas, T., Shieh, J., and Keogh, E. (2010). isax 2.0: Indexing and mining one billion time series. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 58–67. IEEE.

Carpenter, J. R., Kenward, M. G., and Vansteelandt, S. (2006). A comparison of multiple imputation and doubly robust estimation for analyses with missing data. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* **169,** 571–584.

Cervantes, S., Samaranch, L., Vidal-Taboada, J. M., Lamet, I., Bullido, M. J., Frank-García, A., Coria, F., Lleó, A., Clarimón, J., Lorenzo, E., et al. (2011). Genetic variation in apoe cluster region and alzheimer's disease risk. *Neurobiology of Aging* **32,** 2107–e7.

Chambers, R. L., Steel, D. G., Wang, S., and Welsh, A. (2012). *Maximum likelihood estimation for sample surveys*. Chapman and Hall/CRC.

Chan, K.-P. and Fu, W.-C. (1999). Efficient time series matching by wavelets. In *icde*, page 126. IEEE.

Chang, T. and Kott, P. S. (2008). Using calibration weighting to adjust for nonresponse under a plausible model. *Biometrika* **95,** 555–571.

Chatterjee, N. and Carroll, R. (2005). Semiparametric maximum likelihood estimation in case-control studies of gene-environment interactions. *Biometrika* **92,** 399–418.

Chen, K. (2001). Parametric models for response-biased sampling. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63,** 775–789.

Cheng, X., Zhang, R., Zhou, J., and Xu, W. (2018). Deeptransport: Learning spatial-temporal dependency for traffic condition forecasting. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Chollet, F. et al. (2015). Keras.

Chung, F. R. and Graham, F. C. (1997). *Spectral graph theory*, volume 92. American Mathematical Soc.

Cornfield, J. (1951). A method of estimating comparative rates from clinical data. applications to cancer of the lung, breast, and cervix. *Journal of the National Cancer Institute* **11,** 1269–1275.

Cox, D. R. (1972). Regression models and life-tables. *Journal of Royal Statistical Society* **34,** 187–220.

Cressie, N. and Wikle, C. K. (2015). *Statistics for spatio-temporal data*. John Wiley & Sons.

Cruchaga, C., Kauwe, J. S., Harari, O., Jin, S. C., Cai, Y., Karch, C. M., Benitez, B. A., Jeng, A. T., Skorupa, T., Carrell, D., et al. (2013). Gwas of cerebrospinal fluid tau levels identifies risk variants for alzheimer's disease. *Neuron* **78,** 256–268.

Cui, Z., Ke, R., and Wang, Y. (2016). Deep stacked bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction. In *6th International Workshop on Urban Computing (UrbComp 2017)*.

Daniels, M. J. and Hogan, J. W. (2008). *Missing data in longitudinal studies: Strategies for Bayesian modeling and sensitivity analysis*. Chapman and Hall/CRC.

Das, G., Gunopulos, D., and Mannila, H. (1997). Finding similar time series. In *European Symposium on Principles of Data Mining and Knowledge Discovery*, pages 88–100. Springer.

Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pages 3844–3852.

Deng, D., Shahabi, C., Demiryurek, U., Zhu, L., Yu, R., and Liu, Y. (2016). Latent space model for road networks to predict time-varying traffic. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1525–1534. ACM.

Deville, J.-C. (2000). Generalized calibration and application to weighting for non-response. In *COMPSTAT*, pages 65–76. Springer.

Diggle, P. and Kenward, M. G. (1994). Informative drop-out in longitudinal data analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **43,** 49–73.

Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. (2015). Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232.

Egner, J. R. (2010). Ajcc cancer staging manual. *JAMA* **304,** 1726–1727.

Figalli, A. (2010). The optimal partial transport problem. *Archive for rational mechanics and analysis* **195,** 533–560.

Flanders, W. D. and Greenland, S. (1991). Analytic methods for two-stage case-control studies and other stratified designs. *Statistics in Medicine* **10,** 739–747.

Ge, X. and Smyth, P. (2000). Deformable markov model templates for time-series pattern matching. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 81–90. ACM.

Glynn, R. J., Laird, N. M., and Rubin, D. B. (1993). Multiple imputation in mixture models for nonignorable nonresponse with follow-ups. *Journal of the American Statistical Association* **88,** 984–993.

Greenlees, J. S., Reece, W. S., and Zieschang, K. D. (1982). Imputation of missing values when the probability of response depends on the variable being imputed. *Journal of the American Statistical Association* **77,** 251–261.

Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM.

Hammond, D. K., Vandergheynst, P., and Gribonval, R. (2011). Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis* **30,** 129–150.

Hao, Y., Wang, T., Zhang, X., Duan, Y., Yu, C., Jiang, T., Fan, Y., and Initiative, A. D. N. (2014). Local label learning (lll) for subcortical structure segmentation: application to hippocampus segmentation. *Human Brain Mapping* **35,** 2674–2697.

He, J., Li, H., Edmondson, A. C., Rader, D. J., and Li, M. (2012). A gaussian copula approach for the analysis of secondary phenotypes in case–control genetic association studies. *Biostatistics* **13,** 497–508.

Heckman, J. J. (1976). The common structure of statistical models of truncation, sample selection and limited dependent variables and a simple estimator for such models. In *Annals of Economic and Social Measurement, Volume 5, number 4*, pages 475–492. NBER.

Hellinger, E. (1909). Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik* **136,** 210–271.

Henaff, M., Bruna, J., and LeCun, Y. (2015). Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163* .

Ibrahim, J. G., Chen, M.-H., Lipsitz, S. R., and Herring, A. H. (2005). Missing-data methods for generalized linear models: A comparative review. *Journal of the American Statistical Association* **100,** 332–346.

Ibrahim, J. G. and Lipsitz, S. R. (1996). Parameter estimation from incomplete data in binomial regression when the missing data mechanism is nonignorable. *Biometrics* pages 1071–1078.

Ibrahim, J. G., Lipsitz, S. R., and Chen, M.-H. (1999). Missing covariates in generalized linear models when the missing data mechanism is non-ignorable. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **61,** 173–190.

Idé, T. and Sugiyama, M. (2011). Trajectory regression on road networks. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Jack Jr, C. R., Bernstein, M. A., Fox, N. C., Thompson, P., Alexander, G., Harvey, D., Borowski, B., Britson, P. J., L. Whitwell, J., Ward, C., et al. (2008). The alzheimer's disease neuroimaging initiative (adni): Mri methods. *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine* **27,** 685–691.

Jiang, Y., Scott, A. J., and Wild, C. J. (2006). Secondary analysis of case-control data. *Statistics in Medicine* **25,** 1323–1339.

Karpatne, A., Ebert-Uphoff, I., Ravela, S., Babaie, H. A., and Kumar, V. (2018). Machine learning for the geosciences: Challenges and opportunities. *IEEE Transactions on Knowledge and Data Engineering* .

Kawahara, J., Brown, C. J., Miller, S. P., Booth, B. G., Chau, V., Grunau, R. E., Zwicker, J. G., and Hamarneh, G. (2017). Brainnetcnn: convolutional neural networks for brain networks; towards predicting neurodevelopment. *Neuroimage* **146,** 1038–1049.

Ke, J., Zheng, H., Yang, H., and Chen, X. M. (2017). Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. *Transportation Research Part C: Emerging Technologies* **85,** 591–608.

Kim, D. H., Payne, M. E., Levy, R. M., MacFall, J. R., and Steffens, D. C. (2002). Apoe genotype and hippocampal volume change in geriatric depression. *Biological Psychiatry* **51,** 426–429.

Kim, J., Pan, W., Initiative, A. D. N., et al. (2015). A cautionary note on using secondary phenotypes in neuroimaging genetic studies. *Neuroimage* **121,** 136–145.

Kim, J. K. and Yu, C. L. (2011). A semiparametric estimation of mean functionals with nonignorable missing data. *Journal of the American Statistical Association* **106,** 157–165.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* .

Korn, F., Jagadish, H. V., and Faloutsos, C. (1997). Efficiently supporting ad hoc queries in large datasets of time sequences. In *Acm Sigmod Record*, volume 26, pages 289–300. ACM.

Kwon, J. and Murphy, K. (2000). Modeling freeway traffic with coupled hmms. Technical report, Technical report, Univ. California, Berkeley.

Lee, A., McMurchy, L., and Scott, A. (1997). Re-using data from case-control studies. *Statistics in Medicine* **16,** 1377–1389.

Li, X., Pan, G., Wu, Z., Qi, G., Li, S., Zhang, D., Zhang, W., and Wang, Z. (2012). Prediction of urban human mobility using large-scale taxi traces and its applications. *Frontiers of Computer Science* **6,** 111–121.

Li, Y., Tarlow, D., Brockschmidt, M., and Zemel, R. (2015). Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493* .

Li, Y., Yu, R., Shahabi, C., and Liu, Y. (2018). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting.

Lin, D. Y. and Zeng, D. (2009). Proper analysis of secondary phenotype data in case-control association studies. *Genetic Epidemiology* **33,** 256–265.

Lippi, M., Bertini, M., and Frasconi, P. (2013). Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *IEEE Transactions on Intelligent Transportation Systems* **14,** 871–882.

Little, R. J. (1994). A class of pattern-mixture models for normal incomplete data. *Biometrika* **81,** 471–483.

Little, R. J. and Rubin, D. B. (2019). *Statistical analysis with missing data*, volume 793. Wiley.

Liu, E. Y., Li, M., Wang, W., and Li, Y. (2013). Mach-admix: genotype imputation for admixed populations. *Genetic Epidemiology* **37,** 25–37.

Lu, P. H., Thompson, P. M., Leow, A., Lee, G. J., Lee, A., Yanovsky, I., Parikshak, N., Khoo, T., Wu, S., Geschwind, D., et al. (2011). Apolipoprotein e genotype is associated with temporal and hippocampal atrophy rates in healthy elderly adults: a tensor-based morphometry study. *Journal of Alzheimer's Disease* **23,** 433–442.

Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., and Wang, Y. (2017). Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. *Sensors* **17,** 818.

Ma, Y. and Carroll, R. J. (2016). Semiparametric estimation in the secondary analysis of case–control studies. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **78,** 127–151.

Ma, Y. et al. (2010). A semiparametric efficient estimator in case-control studies. *Bernoulli* **16,** 585–603.

Manessi, F., Rozza, A., and Manzo, M. (2017). Dynamic graph convolutional networks. *arXiv preprint arXiv:1704.06199* .

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Molenberghs, G. and Kenward, M. (2007). *Missing data in clinical studies*, volume 61. John Wiley & Sons.

Monsees, G. M., Tamimi, R. M., and Kraft, P. (2009). Genome-wide association scans for secondary traits using case-control samples. *Genetic Epidemiology* **33,** 717–728.

Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., and Damas, L. (2013). Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems* **14,** 1393–1402.

Pan, B., Demiryurek, U., and Shahabi, C. (2012). Utilizing real-world transportation data for accurate traffic prediction. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 595–604. IEEE.

Peress, M. (2010). Correcting for survey nonresponse using variable response propensity. *Journal of the American Statistical Association* **105,** 1418–1430.

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM.

Piccoli, B. and Rossi, F. (2014). Generalized wasserstein distance and its application to transport equations with source. *Archive for Rational Mechanics and Analysis* **211,** 335–358.

Potkin, S. G., Macciardi, F., Guffanti, G., Fallon, J. H., Wang, Q., Turner, J. A., Lakatos, A., Miles, M. F., Lander, A., Vawter, M. P., and X.Xie (2010). Identifying gene regulatory networks in schizophrenia. *Neuroimage* **53,** 839–847.

Prentice, R. L. and Pyke, R. (1979). Logistic disease incidence models and case-control studies. *Biometrika* **66,** 403–411.

Price, A. L., Patterson, N. J., Plenge, R. M., Weinblatt, M. E., Shadick, N. A., and Reich, D. (2006). Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics* **38,** 904.

Qin, J., Leung, D., and Shao, J. (2002). Estimation with survey data under nonignorable nonresponse or informative sampling. *Journal of the American Statistical Association* **97,** 193–200.

Richardson, D. B., Rzehak, P., Klenk, J., and Weiland, S. K. (2007). Analyses of case–control data for additional outcomes. *Epidemiology* **18,** 441–445.

Robins, J. M., Rotnitzky, A., and Zhao, L. P. (1994). Estimation of regression coefficients when some regressors are not always observed. *Journal of the American statistical Association* **89,** 846–866.

Robins, J. M., Rotnitzky, A., and Zhao, L. P. (1995). Analysis of semiparametric regression models for repeated outcomes in the presence of missing data. *Journal of the American Statistical Association* **90,** 106–121.

Rong, L., Cheng, H., and Wang, J. (2017). Taxi call prediction for online taxicab platforms. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint Conference on Web and Big Data*, pages 214–224. Springer.

Rosenbaum, P. R. and Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika* **70,** 41–55.

Rosset, S., Zhu, J., Zou, H., and Hastie, T. J. (2005). A method for inferring label sampling mechanisms in semi-supervised learning. In *Advances in neural information processing systems*, pages 1161–1168.

Rubin, D. B. (1976). Inference and missing data. *Biometrika* **63,** 581–592.

Rubin, D. B. (1996). Multiple imputation after 18+ years. *Journal of the American statistical Association* **91,** 473–489.

Rubin, D. B. (2004). *Multiple imputation for nonresponse in surveys*, volume 81. John Wiley & Sons.

Schafer, J. L. (1997). *Analysis of incomplete multivariate data*. Chapman and Hall/CRC.

Schafer, J. L. and Schenker, N. (2000). Inference with imputed conditional means. *Journal of the American Statistical Association* **95,** 144–154.

Schifano, E. D., Li, L., Christiani, D. C., and Lin, X. (2013). Genome-wide association analysis for multiple continuous secondary phenotypes. *The American Journal of Human Genetics* **92,** 744–759.

Schuff, N., Woerner, N., Boreta, L., Kornfield, T., Shaw, L., Trojanowski, J., Thompson, P., Jack Jr, C., Weiner, M., and Initiative, A. D. N. (2009). Mri of hippocampal volume loss in early alzheimer's disease in relation to apoe genotype and biomarkers. *Brain* **132,** 1067–1077.

Seo, Y., Defferrard, M., Vandergheynst, P., and Bresson, X. (2018). Structured sequence modeling with graph convolutional recurrent networks. In *International Conference on Neural Information Processing*, pages 362–373. Springer.

Shao, J. and Wang, L. (2016). Semiparametric inverse propensity weighting for nonignorable missing data. *Biometrika* **103,** 175–187.

Shekhar, S. and Williams, B. (2008). Adaptive seasonal time series models for forecasting short-term traffic flow. *Transportation Research Record: Journal of the Transportation Research Board* pages 116–125.

Shen, L., Kim, S., Risacher, S. L., Nho, K., Swaminathan, S., West, J. D., Foroud, T., Pankratz, N., Moore, J. H., Sloan, C. D., et al. (2010). Whole genome association study of brain-wide imaging phenotypes for identifying quantitative trait loci in mci and ad: A study of the adni cohort. *Neuroimage* **53,** 1051–1063.

Sikov, A. (2018). A brief review of approaches to non-ignorable non-response. *International Statistical Review* **86,** 415–441.

Sofer, T., Cornelis, M. C., Kraft, P., and Tchetgen, E. T. (2017). Control function assisted ipw estimation with a secondary outcome in case-control studies. *Statistica Sinica* **27,** 785–804.

Song, X., Ionita-Laza, I., Liu, M., Reibman, J., and We, Y. (2016). A general and robust framework for secondary traits analysis. *Genetics* pages genetics–115.

Tang, G., Little, R. J., and Raghunathan, T. E. (2003). Analysis of multivariate missing data with nonignorable nonresponse. *Biometrika* **90,** 747–764.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. (2015). Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. International World Wide Web Conferences Steering Committee.

Tang, N., Zhao, P., and Zhu, H. (2014). Empirical likelihood for estimating equations with nonignorably missing data. *Statistica Sinica* **24,** 723.

Tchetgen Tchetgen, E. (2014). A general regression framework for a secondary outcome in case-control studies. *Biostatistics* **5,** 117–128.

Thies, W. and Bleiler, L. (2012). 2012 alzheimer's disease facts and figures. *Alzheimer's & dementia: the Journal of the Alzheimer's Association* .

Thijs, H., Molenberghs, G., Michiels, B., Verbeke, G., and Curran, D. (2002). Strategies to fit pattern-mixture models. *Biostatistics* **3,** 245–265.

Villani, C. (2008). *Optimal transport: old and new*, volume 338. Springer Science & Business Media.

Wang, C.-W., Lee, Y.-C., Calista, E., Zhou, F., Zhu, H., Suzuki, R., Komura, D., Ishikawa, S., and Cheng, S.-P. (2017). A benchmark for comparing precision medicine methods in thyroid cancer diagnosis using tissue microarrays. *Bioinformatics* **34,** 1767–1773.

Wang, D., Cao, W., Li, J., and Ye, J. (2017). Deepsd: supply-demand prediction for online car-hailing services using deep neural networks. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 243–254. IEEE.

Wang, S., Shao, J., and Kim, J. K. (2014). An instrumental variable approach for identification and estimation with nonignorable nonresponse. *Statistica Sinica* pages 1097–1116.

Wei, H., Wang, Y., Wo, T., Liu, Y., and Xu, J. (2016). Zest: a hybrid model on predicting passenger demand for chauffeured car service. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 2203–2208. ACM.

Wei, J., Carroll, R. J., Müller, U. U., Keilegom, I. V., and Chatterjee, N. (2013). Robust estimation for homoscedastic regression in the secondary analysis of case–control data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **75,** 185–206.

Wei, L. and Keogh, E. (2006). Semi-supervised time series classification. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 748–753. ACM.

Weston, J., Ratle, F., Mobahi, H., and Collobert, R. (2012). Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, pages 639–655. Springer.

White, J. E. (1982). A two stage design for the study of the relationship between a rare exposure and a rare disease. *American Journal of Epidemiology* **115,** 119–128.

Wu, F., Wang, H., and Li, Z. (2016). Interpreting traffic dynamics using ubiquitous urban data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 69. ACM.

Wu, Y.-L., Agrawal, D., and El Abbadi, A. (2000). A comparison of dft and dwt based similarity search in time-series databases. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 488–495. ACM.

Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. (2015). Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810.

Yan, S., Xiong, Y., and Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:1801.07455* .

Yang, B., Guo, C., and Jensen, C. S. (2013). Travel cost inference from sparse, spatio temporally correlated time series using markov models. *Proceedings of the VLDB Endowment* **6,** 769–780.

Yang, F., Lorch, S. A., Small, D. S., et al. (2014). Estimation of causal effects using instrumental variables with nonignorable missing covariates: Application to effect of type of delivery nicu on premature infants. *The Annals of Applied Statistics* **8,** 48–73.

Yang, K. and Shahabi, C. (2004). A pca-based similarity measure for multivariate time series. In *Proceedings of the 2nd ACM international workshop on Multimedia databases*, pages 65–74. ACM.

Yang, Z., Cohen, W. W., and Salakhutdinov, R. (2016). Revisiting semi-supervised learning with graph embeddings. *arXiv preprint arXiv:1603.08861* .

Yao, H., Tang, X., Wei, H., Zheng, G., Yu, Y., and Li, Z. (2018). Modeling spatial-temporal dynamics for traffic prediction. *arXiv preprint arXiv:1803.01254* .

Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., Gong, P., Ye, J., and Li, Z. (2018). Deep multi-view spatial-temporal network for taxi demand prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Yu, B., Yin, H., and Zhu, Z. (2018). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *in IJCAI 2018* .

Yu, H., Wu, Z., Wang, S., Wang, Y., and Ma, X. (2017). Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. *Sensors* **17,** 1501.

Yu, R., Li, Y., Shahabi, C., Demiryurek, U., and Liu, Y. (2017). Deep learning: A generic approach for extreme condition traffic forecasting. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 777–785. SIAM.

Zakaria, J., Mueen, A., and Keogh, E. (2012). Clustering time series using unsupervised-shapelets. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 785–794. IEEE.

Zhang, J., Zheng, Y., and Qi, D. (2017). Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Zhang, J., Zheng, Y., Qi, D., Li, R., and Yi, X. (2016). Dnn-based prediction model for spatio-temporal data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 92. ACM.

Zhao, H., Zhao, P.-Y., and Tang, N.-S. (2013). Empirical likelihood inference for mean functionals with nonignorably missing response data. *Computational Statistics & Data Analysis* **66,** 101–116.

Zhao, J. and Shao, J. (2015). Semiparametric pseudo-likelihoods in generalized linear models with nonignorable missing data. *Journal of the American Statistical Association* **110,** 1577–1590.

Zhao, L. and Lipsitz, S. (1992). Designs and analysis of two-stage studies. *Statistics in medicine* **11,** 769–782.

Zheng, J. and Ni, L. M. (2013). Time-dependent trajectory regression on road networks via multi-task learning. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Zhou, X., Shen, Y., Zhu, Y., and Huang, L. (2018). Predicting multi-step citywide passenger demands using attention-based neural networks. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 736–744. ACM.

Zhu, W., Yuan, Y., Zhang, J., Zhou, F., Knickmeyer, R. C., Zhu, H., Initiative, A. D. N., et al. (2017). Genome-wide association analysis of secondary imaging phenotypes from the alzheimer's disease neuroimaging initiative study. *Neuroimage* **146,** 983–1002.

Zhu, X., Ghahramani, Z., and Lafferty, J. D. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pages 912–919.