Statistical Methods for Spatio-Temporal Tensor Data

by

Hu Sun

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Statistics)
in the University of Michigan
2024

Doctoral Committee:

Assistant Professor Yang Chen, Chair
Professor Jian Kang
Research Professor Ward Manchester
Assistant Professor Kean Ming Tan

Hu Sun
husun@umich.edu
ORCID iD: 0000-0002-3110-7041

Dedicated to my father Lifu Sun and my mother Yanhua Zhang.

ACKNOWLEDGEMENTS

This thesis marks the end of my pursuit of the doctoral degree in statistics. As I wrote these words, I sensed an overwhelming emotion, a complex blend of gratitude, fulfillment, promise, and enchantment. I feel very fortunate to have spent my last six years in this beloved college town in Michigan and met all the lovely people who provided me with academic mentorship and emotional support.

First, I would love to express my deepest gratitude and respect to my doctoral advisor, Prof. Yang Chen. I can still remember very clearly the first time we met in the solar flare research group meeting back in January 2019, you provided a lot of support, guidance, and patience for junior researchers like myself. It is your consistent mentorship that enabled me to grow from a junior master's student to a senior doctoral student and made me become a well-equipped statistician. There have been plenty of difficult times during my doctoral degree study, like the COVID-19 era, the job-seeking season, and several paper revision processes, and you've always shown up with support, belief, and encouragement. This thesis would never materialize without you.

In the meantime, I would also love to thank all of my research collaborators. The mentorship from Prof. Shasha Zou, Prof. Ward Manchester, and Prof. Zuofeng Shang has guided me on various difficult but mesmerizing theoretical and application topics that make the backbone of this thesis. I would also express my appreciation to Prof. Yuekai Sun, Prof. Tamas Gombosi, Prof. Alfred Hero, Dr. Xiantong Wang, Dr. Zihan Wang, Dr. Jiaen Ren, Dr. Meng Jin, Dr. Yang Liu, Dr. Yu Wang, Dr. Zeyu Sun, and Zhenbang Jiao, Zhijun Hua and Yurui Chang for the efforts you've spared in our collaborative researches. It is my greatest pleasure to work with all of you.

Apart from the research collaborators, I also want to thank all of the course instructors whom I have had the fortune to work with. I can never emphasize more on how Prof. Ji Zhu has influenced me during the time I served as GSI for STATS 503 on being a responsible and patient educator and statistics practitioner. Also, it was a very memorable experience to work as the GSI for Dr. Mark Fredrickson, Dr. Brandon Legried, and Ms. Nadiya Fink as you all spared substantial efforts to make my job a lot easier.

The pursuit of my doctoral degree is never complete without the accompany of friends in and outside of the department. I would like to thank Zhenbang, Shuoran, Xingbao,

Wang, Yuqing, and Anran for all the relaxing time we spent together during my first two years here as a master's student, which really helps me settle down at a place seven thousand miles away from home. I can never forget Tianci, Songkai, and Gang for always being supportive friends both academically and in life. I appreciate the help and encouragement from Bo, Jinming, Ziping, Yang, and Yuanzhi during my internship search. I want to thank Daniele for inviting me to perform during my last statsgiving and Kevin, our wonder guitarist. I want to thank Gabriel, Felipe, Eduardo, and Alexander for inviting me to multiple international student parties that lightened up my summer in Ann Arbor. I also want to thank Shuqing and Jukai who kindly served as my barber and had lots of wonderful dinners with me during the darkest times of COVID. Finally, I want to thank Wayne, Rob, Daniel, Bach, Victor, Noah, Yuxuan, Jiuqian, Kevin, Ashlan, Soham, Jingyang, and Yiluan from our lovely lab who always encouraged me and made me feel comfortable and confident for research talks and our group events.

The dissertation defense is not possible without Prof. Jian Kang, Prof. Kean Ming Tan, and Prof. Ward Manchester who kindly agreed to serve on my committee and provided me with abundant suggestions on research and career development. It is also not possible without my undergraduate advisor, Prof. Yun Wang, who opened up the world of scientific research for me.

Lastly, I would love to show my deepest gratitude and love to my family and loved ones. Words elude me when I articulate my appreciation for my father Lifu Sun and my mother Yanhua Zhang. You have supported me academically, financially, and emotionally with unlimited patience and dedication for the past 28 years. It has always been my dream to become a Ph.D. like both of you and my passion never fades only because of you. Finally, I want to say thank you to my beautiful, smart, and considerate girlfriend, Shushu. Meeting you in this chilly, snowy college town is like a fairy tale to me and you add warmth and vibrant colors to my life that make these times unforgettable. This thesis is also a love letter as I wrote every page with you by my side.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

LIST OF APPENDICES

ABSTRACT

In recent years, tensors have garnered significant attention from researchers across the domains of statistics, applied mathematics, and machine learning. The inherent multi-linear structure of tensors renders them an efficient means of representing high-dimensional data. The technological revolution in data collection and processing has led to the emergence of tensorial datasets across numerous scientific applications, such as neuroimaging, collaborative filtering, and longitudinal data analysis. In this thesis, we focus specifically on the analysis of tensors with spatial and temporal dimensionality, commonly referred to as spatio-temporal tensors. We leverage the efficient tensor representation to analyze large-scale spatio-temporal data and integrate intricate spatio-temporal dependencies into the tensor model. Inspired by scientific applications in space weather monitoring, we introduce novel statistical methodologies addressing four distinct challenges.

I) The first part investigates the missing value imputation of spatio-temporal tensors with locally dependent missingness. Traditional low-rank matrix/tensor completion methods cannot provide reasonable imputations at locations where almost all data are missing in the neighborhood. We adopt the classic low-rank matrix completion framework and improve it by giving a tensor completion estimator exhibiting spatial and temporal continuity. We establish the convergence guarantee of the new method and apply it extensively to the global Total Electron Content (TEC) reconstruction problem.

II) The second part dives into the uncertainty quantification (UQ) of tensor completion. Literature on the UQ of tensor completion relies heavily on the assumption that data is missing uniformly at random or at least independently and only applies to a restricted class of the completion method. We circumvent these restrictions by introducing a conformal prediction framework for the UQ. The resulting confidence intervals are constructed by properly accounting for the missing propensity of each tensor entry, which is estimated by a low-rank tensor Ising model that can account for the dependent data missingness. We establish the theoretical coverage guarantee and validate the method through extensive simulations and an application to the global TEC reconstruction problem.

III) The third part focuses on the forecasting problem of matrix-valued spatial time series with auxiliary vector-valued, non-spatial time series covariates. Existing works on

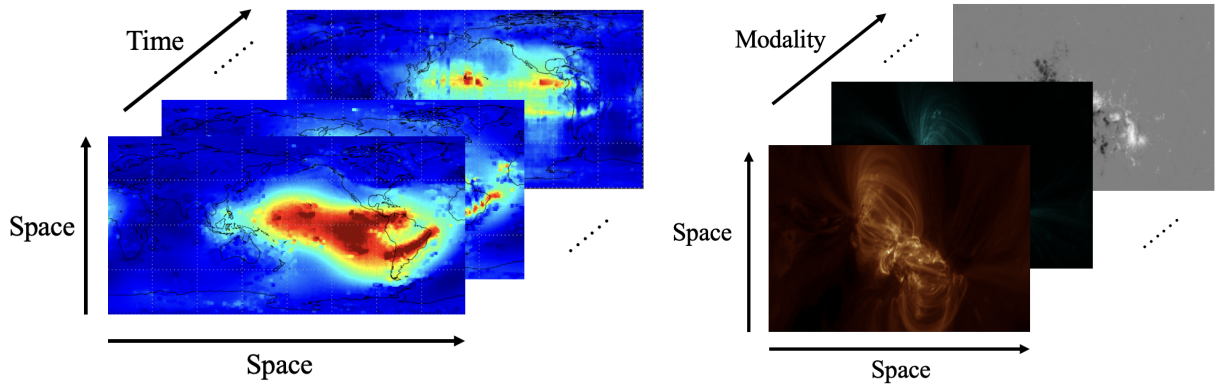
matrix autoregression cannot handle such settings with predictors of non-uniform modes and spatio-temporal dimensions. We propose a novel semi-parametric matrix autoregression model incorporating the vector covariates with spatially smooth tensor coefficients. We establish the joint asymptotics of the autoregressive and tensor parameters under fixed and high-dimensional regimes and apply our method to the global TEC forecast problem.

IV) The last part is dedicated to a scalar-on-tensor regression problem with multi-modal imaging tensor covariate. We encapsulate a tensor dimension reduction step and a Gaussian Process regression model in a single framework and introduce a total-variation regularization to capture spatially contiguous predictive signals. The new model complements the current literature by accounting for the interplay among different data modalities in an interpretable fashion. We apply our model to forecast the intensity of solar flares with multi-channel solar imaging data.

CHAPTER 1

Introduction

In this thesis, we present analyses of tensor data with spatial dimensionality. Such type of data contains the auxiliary information indicating where the data was collected and all observations are aligned in a multi-dimensional array. To facilitate the discussions, we provide two concrete examples of such types of data. In Figure 1.1a, we give a schematic illustration of *spatio-temporal tensor data*. In this example, each data point $x_{\mathbf{i}}$ lies on a 3-dimensional tensor grid and can be uniquely identified by a 3-tuple $\mathbf{i} = (i_1, i_2, i_3)$, where i_1, i_2 are indices for the spatial dimensions and i_3 is the index for the temporal dimension. In Figure 1.1b, we visualize a similar data format called *multi-modal spatial data*, which differs from the spatio-temporal tensor data only in that one replaces the temporal dimension with the data modality dimension. Specifically in Figure 1.1b, the data modality is the different imaging channels.



(a) Spatio-Temporal Tensor Data (b) Multi-Modality Spatial Data

Figure 1.1: Examples of tensor data with spatial dimensionality.

Modern technological society has witnessed a burst of such large-scale spatial tensor datasets in multiple scientific fields, such as spatio-temporal tensor data in remote sensing (Shen et al., 2016; Li et al., 2023), climatology (Lozano et al., 2009; Bahadori et al., 2014;

Yu and Liu, 2016), economics (Zhang et al., 2017) and epidemiology (Kargas et al., 2021), multi-modality spatial data in neuroimaging (Sui et al., 2012; Karahan et al., 2015) and astrophysics (Jonas et al., 2018; Sun et al., 2023b). This thesis focuses on the analyses of both spatio-temporal tensor data (Chapters 2, 3, 4) and multi-modality spatial data (Chapter 5). In the following discussions, we will use spatio-temporal tensor data to illustrate our motivations and the arguments naturally apply to multi-modality spatial data.

In the classical setting of spatial-temporal statistics (Cressie and Wikle, 2015), each observation can be represented as (x_i, s_i, t_i), with $s_i \in \mathbb{R}^d$ being the d-dimensional spatial coordinate and $t_i \in \mathbb{R}$ being the time index. The spatio-temporal tensor data differs from this setting in that each observation can be represented instead as $(x_{i_1,\ldots,i_d,i_{d+1}}, i_1, \ldots, i_d, i_{d+1})$, where $(i_1, \ldots, i_d)$ are the discrete spatial coordinates in a d-dimensional tensor grid and i_{d+1} is the discrete time index. As compared to traditional spatial-temporal data, the tensor representation is more structured and efficient. One can analyze the spatio-temporal tensor data in the classical framework of spatio-temporal statistics by simply vectorizing the tensor data and treating each tensor entry as an individual observation. However, doing so will lose the structural information as well as the efficiency of the tensor representation. Therefore, the need for novel statistical methodologies that can leverage the tensor representation of such spatio-temporal data is in strong demand.

In the classical setting of tensor data analysis (Kolda and Bader, 2009; Cichocki et al., 2015), each entry of a K-mode tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$ is identified by a K-tuple and the ordering of the tensor entries is not based on any notion of adjacency in space or time. For example, in recommender system (Bi et al., 2018), a 3-mode user rating tensor has its three modes being user, item, and context. In a K-mode hypergraph network adjacency tensor (Ke et al., 2019), each mode is the node in the network. One can analyze the spatio-temporal tensor data under the classical tensor learning framework such as low-rank tensor learning (Han et al., 2022), without explicitly dealing with the spatio-temporal dependency of the tensor entries. The notion of tensor rank is not unique, but generally speaking, a low-rank tensor suggests that any matricized version of itself has a low matrix rank. Such low-rank assumption will lose the spatial and temporal information of the data and the failure of capturing the dependency among the data may lead to misleading results in the downstream statistical inference or scientific interpretations of the results. Therefore, there is also a strong demand for novel tensor learning frameworks that can accommodate the spatio-temporal nature of the data.

In this thesis, we make several contributions to developing novel statistical methodologies for spatio-temporal tensor data and multi-modality spatial data. We combine techniques from both spatial(-temporal) statistics and tensor data modeling to devise new sta-

tistical methodologies for four problems: 1) missing value imputation for spatio-temporal tensor data; 2) uncertainty quantification of tensor completion with dependent missingness; 3) forecasting spatio-temporal data with autoregression and non-spatial time series covariates; 4) scalar label regression with multi-modal spatial data.

Each of these four problems has very concrete application contexts. Throughout the thesis, we mainly focus on one spatio-temporal tensor dataset from geophysics and one multi-modal spatial dataset from astrophysics. Both datasets pose new challenges that traditional spatio-temporal statistics or tensor data modeling cannot readily resolve. We now briefly describe the scientific background of both datasets that motivate our methodology research.

1.1 Background of Space Weather Data

"Space Weather" refers to the variable conditions on the Sun and in the near-Earth space environment that can adversely influence the performance and reliability of space-borne and ground-based technological systems (Board, 1997). In recent years, there has been a growing awareness of space weather impacts on critical infrastructure in the civilian, commercial, and military sectors. Understanding the underlying physical processes of space weather and improving forecasting is a major objective of the space science community as well as the focus of applications in this thesis.

In this thesis, we consider applications of spatio-temporal tensor methods to concrete questions in monitoring space weather. We mainly consider two datasets concerning the space weather conditions of the Earth and the Sun. The first dataset is the global total electron content (TEC) data that provides information on the electron density of the Earth's ionosphere. Monitoring the global TEC distribution is critical since ionospheric disturbances can disrupt satellite navigation and communication systems as well as long-distance radio communication. The second dataset is the solar flare data containing both the solar flare event history as well as the solar imaging data describing the photospheric magnetic field and ultraviolet coronal emission. A solar flare is an intense localized eruption of electromagnetic radiation in the Sun's atmosphere. Solar flares with high-energy radiation emissions can strongly impact the Earth's space weather such as the global TEC distribution and potentially interfere with the radio communication of the Earth. This thesis aims to provide a systematic application for analyzing these datasets based on a statistical framework, which can complement existing physical methodologies and provide a unique data-driven perspective. We provide a brief description of each dataset below.

1.1.1 Global Total Electron Content (TEC) Data

The ionosphere, a layer in the upper atmosphere that extends from 70 km to 1000 km above the Earth's surface, contains a roughly equal number of electrons and ions, which are mainly produced by the ionization of the neutral atmosphere by solar UV and EUV radiation and impact ionization by precipitating energetic particles from space. Ionospheric state and variability depend on solar activity, near-Earth space environment conditions, time of day and day of year, as well as the geographic location. Eruptive space weather events, such as coronal mass ejections (CMEs), have the largest impact on ionospheric state and its variability (Mendillo, 2006; Prölss, 2008; Zou et al., 2013b, 2014). Ionospheric disturbances are highlighted as one out of the five major space weather threats in the National Space Weather Strategy and Action Plan (House, 2019).

The Global Navigation Satellite Systems (GNSS) systems were initially designed for Positioning, Navigation, and Timing (PNT) services, but have also been widely used in the space science community for remotely sensing the ionosphere total electron content (TEC). TEC refers to the integrated electron density between the receivers and the GNSS satellites and can be calculated using the different delays of two or more transmitted frequencies from multi-frequency GNSS satellites. To achieve PNT accuracy, single-frequency GNSS receivers on the ground need ionosphere TEC information to remove the ionosphere impact on the speed of propagating radio waves. For example, the Federal Aviation Agency (FAA) developed the Aide Area Augmentation System (WAAS) to estimate and augment the ionosphere delay to improve the PNT service accuracy for aviation. Therefore, specification and forecasting of the ionosphere plasma content, i.e., the TEC map, and its variability are of critical importance to our modern technological society.

The Madrigal Database (Rideout and Coster, 2006; Vierinen et al., 2016) provides global maps of vertical TEC measurements calculated from dual-frequency GNSS data collected by worldwide distributed receivers (over 5000 of them). The GNSS system used here includes both the Global Positioning System (GPS) and the Global Navigation Satellite System (GLONASS). The Madrigal TEC maps are provided with a spatial resolution of 1° latitude by 1° longitude and a temporal resolution of 5 minutes. Mainly due to the absence of GNSS receivers over the oceans, the TEC measurement is missing at around 75% of the globe (see panel (A) of Figure 1.2). In practice, a 3°-by-3° median filter is usually applied to reduce the percentage of missing values down to about 50% (see panel (B) of Figure 1.2).

Complete global ionospheric maps (GIMs) of vertical TEC values provided by the International GNSS Service (IGS) are produced by combining TEC maps calculated by several IGS Ionosphere Associated Analysis Centers (IAACs), which use different techniques but

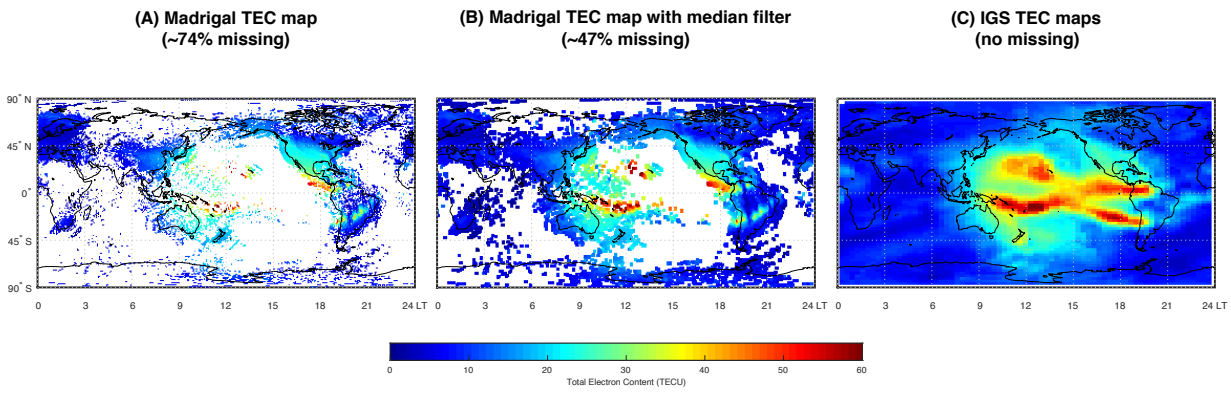


Figure 1.2: TEC map from the Madrigal Database (A) without the median filter on the left, (B) with a $3^\circ \times 3^\circ$ median filter on the right and (C) TEC map from the International GNSS Service (IGS).

mostly expansion in terms of spherical harmonics (SH) to model the global VTEC maps (Schaer, 1999; Hernández-Pajares et al., 2009; Roma-Dollase et al., 2018). The IAACs use TEC data collected by around 200-500 IGS GNSS receivers. The commonly used technique models the global TEC map by an expansion consisting of SH functions, whose coefficients are obtained by fitting the measured TEC data based on the least squared algorithm (Schaer et al., 1995). Additional constraints can be applied to the fitting process to improve the resulting model, e.g., removing negative TEC values by adding an inequality constraint (Zhang et al., 2013). The TEC GIMs provided by IGS have a spatial resolution of 2.5° latitude by 5.0° longitude and the highest temporal resolution of 15 minutes. As shown in panel (C) of Figure 1.2, the IGS TEC maps can well approximate the TEC distribution at a global scale, however, the meso-scale TEC structures are generally smoothed out in these fitted models, while they are important for ionospheric scientific research and for augmenting the ionospheric impact on the PNT service (e.g., Conker et al., 2003; Yang et al., 2020b). For example, in Figures 1.2(A) and 1.2(B), a clear longitudinally extended low TEC channel at about 19 local time (LT), i.e., an equatorial plasma bubble, can be seen clearly, while it is smoothed out in Figure 1.2(C). Equatorial plasma bubble is one of the most important ionospheric density and TEC features that can severely degrade the GPS signals or even lead to loss of lock (Aa et al., 2019; Basu et al., 2002). Therefore, the construction of high spatial resolution TEC data is much needed for both scientific research and operational space weather monitoring and a novel statistical framework that can accomplish this goal is in strong demand.

1.1.2 Solar Flare Data

Solar flares occur in regions of strong magnetic fields possessing concentrated free energy, which are susceptible to spontaneous reconnection that rapidly heats the plasma producing flare emission. At the photosphere, these fields typically take the form of active regions that are characterized by strong horizontal field gradients, long and well-defined polarity-inversion lines (PILs), and complex flux distributions (Falconer et al., 2002, 2003, 2006; Barnes et al., 2007; Schrijver, 2007). In recent years, data-driven flare forecasting has caught much attention in the field of space sciences (Nishizuka et al., 2018; Chen et al., 2019b; Wang et al., 2020; Jiao et al., 2020; Sun et al., 2021, 2022b). Many machine learning algorithms have been adopted for solar flare prediction, either with or without operational forecasting in mind (Barnes et al., 2007). The flare event history data comes from the Geostationary Operational Environmental Satellites (GOES) flare list. There are over $12,000$ solar flares recorded by the Geostationary Operational Environmental Satellite (GOES) from May, 2010 to June, 2017, with intensity at least at the A-class flare level (peak X-ray brightness $< 10^{-7}\mathrm{W/m}^2$). Among these flares, $4,409$ are B-class flares ($10^{-7} \sim 10^{-6}\mathrm{W/m}^2$), 710 are M-class flares ($10^{-5} \sim 10^{-4}\mathrm{W/m}^2$) and 50 are X-class flares ($> 10^{-4}\mathrm{W/m}^2$). The high-intensity event at M-class or above is rare but can have a severe impact on the Earth. In Figure 1.3, we visualize a flare event history for a selected active region on the Sun during a 10-day period in 2011.

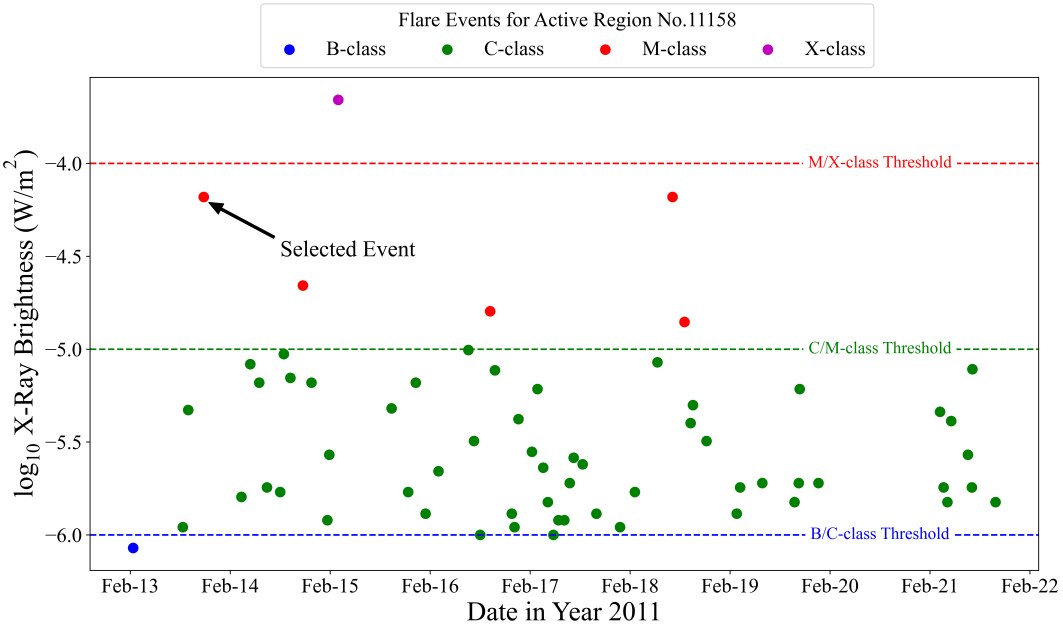


Figure 1.3: Example of solar flare event history for active region No.11158.

Predictors of solar flares include multivariate time-series data in the form of physical parameters (Bobra et al., 2014) or imaging data provided by the Solar Dynamics Observatory (SDO)/Helioseismic and Magnetic Imager (HMI) (Scherrer et al., 2012) and SDO/Atmospheric Imaging Assembly (AIA) (Lemen et al., 2012). For the "selected event" labeled in Figure 1.3, we visualize its corresponding 10-channel solar imaging data from the HMI and AIA database in Figure 1.4.

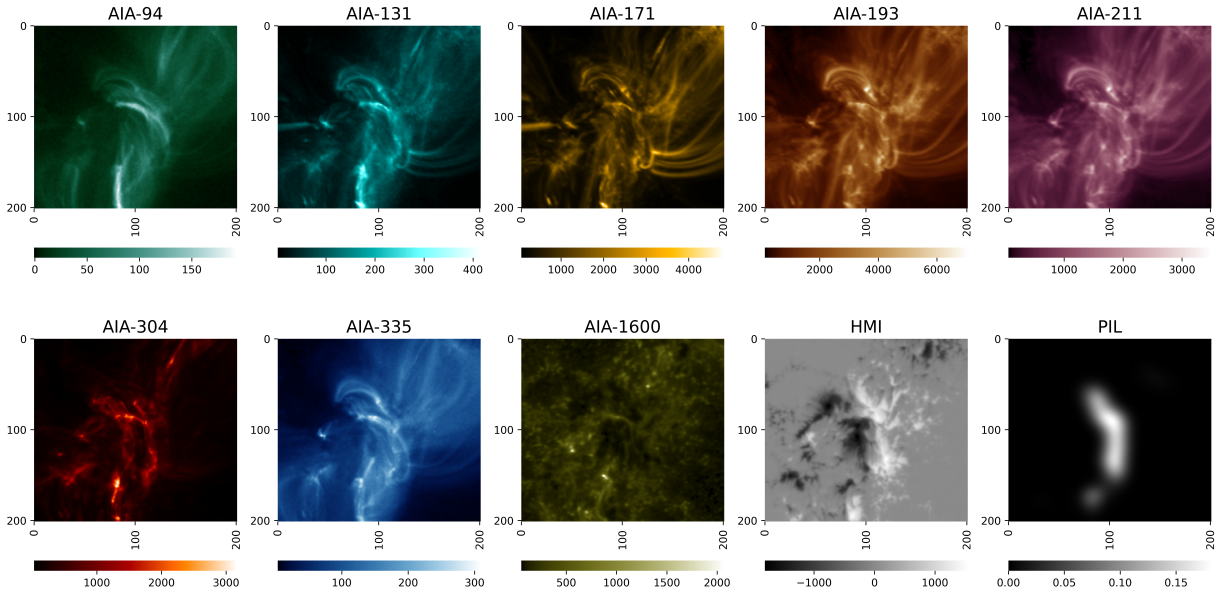


Figure 1.4: Solar imaging data from the HMI and AIA database for the "Selected Event" in Figure 1.3. There are 10 data channels with each channel having a size of 377×744. Channel name is labeled on top of each panel.

Such a dataset poses a significant challenge in a supervised learning setting because one needs to extract flare-discriminating features from ultra high-dimensional predictors with very limited solar flare events. Existing flare prediction models rely on deep learning methods (Chen et al., 2019b; Jiao et al., 2020; Sun et al., 2022b) that are not interpretable for the discovery of new physical signals. Therefore, a novel statistical methodology that can deal with high-dimensional spatial data and detect new physical signals in an interpretable fashion is much needed.

1.2 Organizations & Contributions of the Thesis

The organization of the thesis is summarized graphically in Figure 1.5. There are two major research tracks in the thesis: tensor completion and tensor regression. The former aims at estimating the missing values in the tensor data and the latter conducts supervised

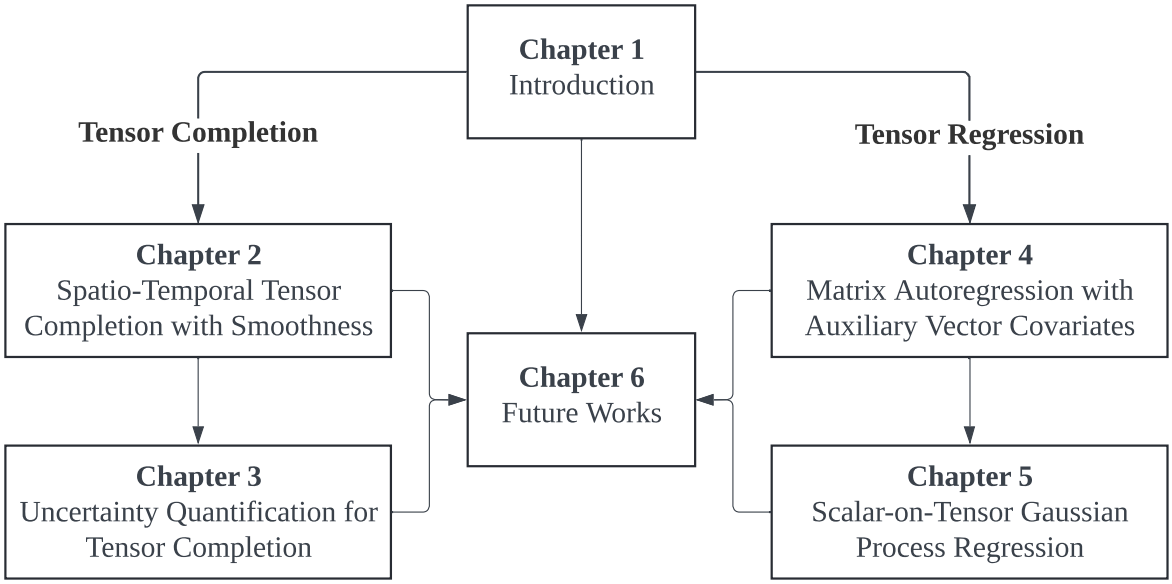


Figure 1.5: The organization of the thesis.

learning with tensor data as covariates. Chapters 2 and 3 deal with tensor completion and Chapters 4 and 5 are dedicated to tensor regression.

In Chapter 2, we propose a methodology for imputing spatio-temporal tensor data that guarantees smoothness across space and time. This chapter is based on our published methodology paper (Sun et al., 2022a) in *Annals of Applied Statistics* and a companion application paper (Sun et al., 2023a) in *Scientific Data*.

In Chapter 3, we further propose a procedure for quantifying the uncertainty of tensor completion estimators using conformal prediction. We investigate the uncertainty quantification problem under data missingness with and without spatial dependency. This chapter is based on our submitted paper *Conformalized Tensor Completion with Riemannian Optimization* (Sun and Chen, 2024).

In Chapter 4, we develop an autoregression model for spatio-temporal tensor data with auxiliary vector time series covariates. We consider forecasting spatio-temporal data using the observed historical spatial data as well as non-spatial data. The framework distinguishes itself from other competing methods by incorporating multi-modal predictors (i.e. both matrices and vectors) in a single framework. This chapter is based on our submitted paper *Matrix Autoregressive Model with Vector Time-Series Covariates for Spatio-Temporal Data* (Sun et al., 2023c).

In Chapter 5, we consider an alternative regression scenario with tensor covariates and scalar responses instead of autoregression. We develop a scalar-on-tensor Gaussian Pro-

cess regression (GPR) model that conducts tensor dimension reduction and tensor regression in a single framework. The tensor dimension reduction step is called tensor contraction and we utilize a total-variation (TV) regularization in this step for extracting predictive signals from the tensor data. This chapter is based on our published paper (Sun et al., 2023b) in *Proceedings of the 40th International Conference on Machine Learning*.

In Chapter 6, we conclude all of the previous chapters and point out future research directions. Figure 1.5 illustrates the relationships among all chapters of the thesis from a statistics methodology perspective. From the application perspective, we can summarize the structure of the thesis under the space weather monitoring context, as depicted in Figure 1.6.

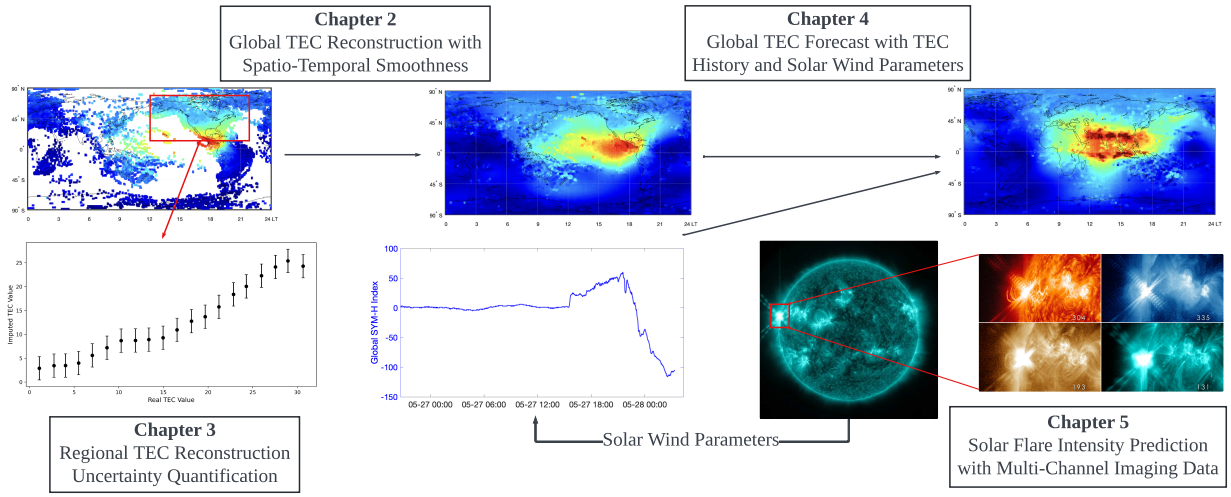


Figure 1.6: Connections among Chapter 2, 3, 4 and 5 through the lens of space weather monitoring applications.

Through the lens of applications to space weather monitoring, Chapter 2 is dedicated to imputing the missing values of the global geomagnetic index, namely the total electron content. Chapter 3 quantifies the uncertainty of the imputed TEC values by constructing confidence regions. Chapter 4 is devoted to making forecasts of future global TEC by utilizing the historical TEC and the solar wind parameter time series. The solar wind parameters measure the strength of the solar wind, which the onset of strong solar flares can heavily impact. Chapter 5 switches the focus from the Earth to the Sun and make predictions on the intensity of solar flares using multi-channel solar imaging data.

This thesis contributes to both the community of statistics and astrophysics/geophysics. For statisticians, we develop a series of methodologies for analyzing spatio-temporal tensor data and provide interpretable, theoretically justifiable, and computationally efficient models for the missing data imputation, autoregression, and scalar-on-tensor re-

gression. As compared to the existing tensor learning literature, our works exploit the spatial and temporal structure of the tensor via matrix factorization with autoregressive structure (Chapter 2), tensor Ising model (Chapter 3), kernel method (Chapter 4) and total-variation regularization (Chapter 5). Each of these techniques augments the existing tensor learning methodologies and makes them adaptive to spatio-temporal data. For astrophysicists and geophysicists, we provide a data-driven framework for analyzing the domain data with interpretability and scalability. We also apply our missing data imputation pipeline to construct an open-access high-resolution global TEC database that covers 16 years to facilitate domain research.

1.3 Notations

The notations used throughout the thesis are consistent. Each chapter might introduce additional notations that are used in that chapter only. For convenience, we summarize all the essential notations shared by all chapters here.

An order-K tensor $\boldsymbol{\mathcal{X}}$ is a multi-dimensional array of size $d_1 \times \cdots \times d_K$. The dimension K of the tensor is commonly referred to as the order, mode, or way of the tensor. We use calligraphic boldface letters such as $\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{G}}$ to denote tensors of order 3 or higher. We use boldface uppercase letters such as $\mathbf{A}, \mathbf{B}$ to denote matrices, which are order-2 tensors. We use boldface lowercase letters such as $\mathbf{z}, \boldsymbol{\beta}$ to denote vectors, which are order-1 tensors. Scalars, which are order-0 tensors, are denoted by plain letters, e.g., c, c_0. Sets are denoted by blackboard boldface letters, e.g., $\mathbb{R}, \mathbb{S}$.

To index a tensor, we use square brackets with subscripts such as $[\boldsymbol{\mathcal{X}}]_{i_1 \ldots i_K}$ and we omit the brackets when it is unambiguous from the context. For a positive integer n, we denote its index set $\{1, \ldots, n\}$ as $[n]$. For any two identical-sized tensors $\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$, we use $\mathbf{vec}(\boldsymbol{\mathcal{X}}), \mathbf{vec}(\boldsymbol{\mathcal{Y}})$ to denote the corresponding vectorized tensors, where all entries are aligned in such an order that the first index changes the fastest. Therefore, $\mathbf{vec}(\mathbf{X})$ is a long vector by stacking columns of the matrix $\mathbf{X}$. We use $\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle$ to denote tensor inner product and basically $\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle = \mathbf{vec}(\boldsymbol{\mathcal{X}})^\top \mathbf{vec}(\boldsymbol{\mathcal{Y}})$. Tensor Frobenius norm $\|\boldsymbol{\mathcal{X}}\|_{\mathrm{F}}$ is defined as $\sqrt{\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}} \rangle}$ and tensor max-norm $\|\boldsymbol{\mathcal{X}}\|_\infty$ is defined as $\max_{i_1, \ldots, i_K} |\boldsymbol{\mathcal{X}}_{i_1, \ldots, i_K}|$.

For any two matrices $\mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{B} \in \mathbb{R}^{p \times q}$, we denote their Kronecker product as $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{mp \times nq}$. We denote the trace of a square matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ as $\mathrm{tr}\,(\mathbf{A})$. For any two vectors $\boldsymbol{\alpha} \in \mathbb{R}^m, \boldsymbol{\beta} \in \mathbb{R}^n$, we denote their outer product as $\boldsymbol{\alpha} \circ \boldsymbol{\beta} \in \mathbb{R}^{m \times n}$.

For any tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$ and any matrix $\mathbf{U} \in \mathbb{R}^{J \times d_k}$, the k-th mode tensor-matrix product, denoted as $\boldsymbol{\mathcal{X}} \times_k \mathbf{U}$, is a tensor of size $d_1 \times \cdots \times d_{k-1} \times J \times d_{k+1} \times \cdots \times d_k$ that

satisfies:

$$[\boldsymbol{\mathcal{X}} \times_k \mathbf{U}]_{i_1 \dots i_{k-1} j i_{k+1} \dots i_K} = \sum_{i_k=1}^{d_k} [\boldsymbol{\mathcal{X}}]_{i_1 \dots i_k \dots i_K} [\mathbf{U}]_{j i_k}.$$

Similarly, for tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{d_1 \times \cdots \times d_{K+1}}$ and vector $\boldsymbol{\beta} \in \mathbb{R}^{d_k}$, the k-th mode tensor-vector product, denoted as $\boldsymbol{\mathcal{X}} \bar{\times}_k \boldsymbol{\beta}$, or simply $\boldsymbol{\mathcal{X}} \bar{\times} \boldsymbol{\beta}$ when $k = K+1$, results in an order-K tensor of size $d_1 \times \cdots \times d_{k-1} \times d_{k+1} \times \cdots \times d_{K+1}$ that satisfies:

$$[\boldsymbol{\mathcal{X}} \bar{\times}_k \boldsymbol{\beta}]_{i_1 \dots i_{k-1} i_{k+1} \dots i_K} = \sum_{i_k=1}^{d_k} [\boldsymbol{\mathcal{X}}]_{i_1 \dots i_k \dots i_{K+1}} [\boldsymbol{\beta}]_{i_k}.$$

Finally, it is useful to define the tensor spectral norm and the tensor nuclear norm here:

Definition 1.3.1. *For a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$, its spectral norm, denoted as $\|\boldsymbol{\mathcal{A}}\|_\sigma$, is defined as:*

$$\|\boldsymbol{\mathcal{A}}\|_\sigma = \sup_{\mathbf{u}_1, \dots, \mathbf{u}_K} \langle \boldsymbol{\mathcal{A}}, \mathbf{u}_1 \circ \cdots \circ \mathbf{u}_K \rangle, \quad \mathbf{u}_k \in \mathbb{S}^{d_k - 1}, \forall k,$$

where $\circ$ denotes vector outer product and $\mathbb{S}^{d_k-1}$ is a unit sphere in $\mathbb{R}^{d_k}$.

Definition 1.3.2. *For a tensor $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$, its nuclear norm $\|\boldsymbol{\mathcal{C}}\|_*$ is defined as:*

$$\|\boldsymbol{\mathcal{C}}\|_* = \inf \left\{ \sum_r \lambda_r \middle| \boldsymbol{\mathcal{C}} = \sum_r \lambda_r \cdot \mathbf{u}_1 \circ \cdots \circ \mathbf{u}_K, \mathbf{u}_k \in \mathbb{S}^{d_k - 1}, \forall k \right\}.$$

More preliminaries on tensor notations and the related algebra will be covered in later chapters and we refer our readers to (Kolda and Bader, 2009) for more references on the related tensor algebra.

CHAPTER 2

Tensor Completion with Spatio-Temporal Smoothness

2.1 Introduction

In this chapter, we consider the problem of imputing the missing values of a 3-D tensor data with two spatial and one temporal mode. Mathematically, the data can be represented by $m \times n$ matrices $\{\mathbf{X}_t, t = 1, 2, \ldots, T\}$, each of which has missing values; and the locations of the missingness vary across different time points. For any arbitrary matrix $\mathbf{X}$, let $\mathbf{\Omega}$ be a binary matrix encoding the observed entries in $\mathbf{X}$; i.e. $[\mathbf{\Omega}]_{ij} = \mathbb{1}_{\{[\mathbf{X}]_{ij} \neq \texttt{NaN}\}}$. Following the notations in Candès and Tao (2010), the projection operator $\mathrm{P}_{\mathbf{\Omega}}(\cdot)$ projects any matrix $\mathbf{X}$ to a matrix $\mathrm{P}_{\mathbf{\Omega}}(\mathbf{X})$ that satisfies:

$$[\mathrm{P}_{\mathbf{\Omega}}(\mathbf{X})]_{ij} = \begin{cases} [\mathbf{X}]_{ij}, & \text{if} [\mathbf{\Omega}]_{ij} = 1, \\ 0, & \text{if} [\mathbf{\Omega}]_{ij} = 0. \end{cases}$$

Similarly, $\mathrm{P}^{\perp}_{\mathbf{\Omega}}(\cdot)$ is an operator that only keeps the values of a matrix where $[\mathbf{\Omega}]_{ij} = 0$ and sets as 0 where $[\mathbf{\Omega}]_{ij} = 1$.

Matrix and tensor completion (Candès and Tao, 2010; Hastie et al., 2015; Yuan and Zhang, 2016; Xia et al., 2021; Cai et al., 2022a) is a technique that provides an estimator of the missing values within a matrix or tensor. Given the aforementioned notations, matrix/tensor completion aims at finding $\widehat{\mathbf{X}}_t$ for $\mathbf{X}_t$ with the goal of making $\mathrm{P}^{\perp}_{\mathbf{\Omega}}(\widehat{\mathbf{X}}_t)$ to be as close as possible to $\mathrm{P}^{\perp}_{\mathbf{\Omega}}(\mathbf{X}^*_t)$, where $\mathbf{X}^*_t$ is the fully-observed matrix at t. In this chapter, we undertake the task of completing the 3-D spatio-temporal tensor by simultaneously completing all $\mathbf{X}_t$ with matrix completion. We guarantee that the completed 3-D tensor is smooth both spatially and temporally to best adapt to the spatio-temporal nature of the data.

As an application of our algorithm, we consider applying our proposed method to the

global total electron content reconstruction problem, as briefly described in Chapter 1.1.1. The objective is to improve upon existing TEC map reconstruction algorithms to obtain maps that comply with the observed values as much as possible while preserving local and global features in the map. The TEC values over the oceans are largely missing due to the lack of GNSS receivers. Therefore, the pattern of missing values is structured: the TEC matrices have big patches of missingness that move across time. Also, the global TEC distribution is smooth across space and time where neighboring entries typically have similar TEC values. Such a database motivates the development of a tensor completion method that can accommodate the spatio-temporal smoothness of the underlying data and handle the dependent data missingness pattern. We will consistently use the TEC map reconstruction problem as an example throughout this chapter to illustrate our motivation as well as methodology.

We will review the literature on the matrix completion problem below. Note that we only discuss the ones that are directly relevant to this chapter and do not give a full literature review on the subject, see Li et al. (2019); Jain et al. (2013); Hastie et al. (2015) and references therein for more related works on low-rank matrix completion. We will cover the literature on tensor completion in Chapter 3.

2.1.1 Literature Review on Matrix Completion

The early matrix completion method with nuclear-norm penalization (Fazel et al., 2001; Fazel, 2002; Mazumder et al., 2010; Recht, 2011) aims at solving the following convex-optimization problem for a partially-observed matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$:

$$\min_{\mathbf{M}} \left\{ H(\mathbf{M}) := \frac{1}{2}\|\mathrm{P}_{\mathbf{\Omega}}(\mathbf{X}-\mathbf{M})\|_{\mathrm{F}}^2 + \lambda\|\mathbf{M}\|_* \right\}, \tag{2.1}$$

where $\|\cdot\|_*$ is the matrix nuclear norm, i.e. sum of all singular values. It is a well-known result that when $\mathbf{X}$ is fully-observed, the solution to (2.1) is $\mathbf{M} = \mathbf{U}_r \boldsymbol{S}_\lambda(\mathbf{D}_r)\mathbf{V}_r^T$, where $r = \min(m,n)$; $\mathbf{U}_r, \mathbf{D}_r, \mathbf{V}_r$ are from the rank-r Singular Value Decomposition (SVD) of $\mathbf{X}$ with $\mathbf{D}_r = \mathbf{diag}\left[(\sigma_1, \sigma_2, \ldots, \sigma_r)\right]$; and $\boldsymbol{S}_\lambda(\mathbf{D}_r) = \mathbf{diag}\left[(\sigma_1-\lambda)_+, (\sigma_2-\lambda)_+, \ldots, (\sigma_r-\lambda)_+\right]$ is the soft-thresholded singular value matrix. And when $\mathbf{X}$ contains missing values, the problem in (2.1) can be solved by an iterative algorithm. In each step of the algorithm, the missing entries in the matrix are first imputed by the current imputation matrix $\widetilde{\mathbf{M}}$ to get an "imputed" $\widetilde{\mathbf{X}}$ and then the soft-thresholded rank-r SVD is done on $\widetilde{\mathbf{X}}$ to update $\widetilde{\mathbf{M}}$.

In Hastie et al. (2015), inspired by the maximum-margin matrix factorization (MMMF) formulation in Srebro et al. (2005), a different formulation called the softImpute-ALS,

which our method is based on, is proposed. It seeks to find factor matrices $\mathbf{A} \in \mathbb{R}^{m\times r}$ and $\mathbf{B} \in \mathbb{R}^{n\times r}$ such that $\mathbf{AB}^\top$ approximates the matrix $\mathbf{X}$. In each iteration of softImpute-ALS, the following optimization problem is solved:

$$\min_{\mathbf{A},\mathbf{B}} \left\{ F(\mathbf{A},\mathbf{B}|\widetilde{\mathbf{A}},\widetilde{\mathbf{B}}) := \frac{1}{2}\|\widehat{\mathbf{X}} - \mathbf{AB}^\top\|_{\mathrm{F}}^2 + \frac{\lambda_1}{2}(\|\mathbf{A}\|_{\mathrm{F}}^2 + \|\mathbf{B}\|_{\mathrm{F}}^2) \right\}, \tag{2.2}$$

where $\|\cdot\|_{\mathrm{F}}$ is the matrix Frobenius norm, defined as the square root of the sum of the squares of all matrix elements, $\widehat{\mathbf{X}}$ is a "filled-in" matrix, with $\widehat{\mathbf{X}} = \mathrm{P}_{\mathbf{\Omega}}(\mathbf{X}) + \mathrm{P}_{\mathbf{\Omega}^\perp}(\widetilde{\mathbf{A}}\widetilde{\mathbf{B}}^\top)$, and $\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}$ are the factor matrices in the previous iteration. The algorithm works via doing alternating ridge regression to update $\mathbf{A}$ and $\mathbf{B}$, and eventually, the solution is $\widehat{\mathbf{A}} = \mathbf{U}_r \boldsymbol{S}_\lambda(\mathbf{D}_r)^{\frac{1}{2}}$ and $\widehat{\mathbf{B}} = \mathbf{V}_r \boldsymbol{S}_\lambda(\mathbf{D}_r)^{\frac{1}{2}}$, which agrees with the solution of (2.1) as long as r is specified to be larger than the rank of the solution of (2.1) (Mazumder et al., 2010).

The nuclear-norm penalty term in (2.1) and the ℓ_2-penalty in (2.2) share a common modeling intention: put a soft constraint on the rank of the imputation matrix, so that the optimization problem is made (bi-)convex. Lemma 1 in Srebro et al. (2005) also shows that $\|\mathbf{X}\|_* = \min_{\mathbf{X}=\mathbf{AB}^\top} \frac{1}{2}(\|\mathbf{A}\|_{\mathrm{F}}^2 + \|\mathbf{B}\|_{\mathrm{F}}^2)$, so the penalty term in (2.2) is a further relaxation of the nuclear-norm penalty. The $\mathbf{A}_{m\times r}$ factor and the $\mathbf{B}_{n\times r}$ factor have direct interpretations: the r-dimensional latent feature for every row and column of the matrix. Each entry in the final imputation is just the inner product of the latent features of the corresponding row and column (e.g., latitude and longitude in the TEC map). We adopt the factorization setup and the ℓ_2-penalty in (2.2) because of its interpretability, direct control of r, and its elegant least-square solution. The default of our algorithm in the following text fixes $r = \min(m, n)$. However, one can impose a low-rank structure in the first place if prior knowledge is present. See Section 8 of Mazumder et al. (2010) for further comparison of the two types of penalty terms.

In our TEC map reconstruction example, one can directly apply the softImpute-ALS to each TEC matrix $\mathbf{X}_t$ separately by iteratively solving (2.2). In Figure 2.1, we show the observed TEC map on the left and the reconstructed TEC map with the fine-tuned softImpute-ALS method on the right. This figure reveals two problems of applying the original softImpute-ALS algorithm directly to the TEC map reconstruction problem and similar spatio-temporal data:

1. The imputed TEC map, namely $\widehat{\mathbf{A}}\widehat{\mathbf{B}}^T$, exhibits a *global* unsmooth structure which is not ideal. The maximum possible rank is 181 for the map but the imputed one generally has rank $70 \sim 90$. There are many rows (or columns) imputed with equal, near-zero values, i.e., the blue bands in the right panel of Figure 2.1 at about 15 MLT[1] and

[1]MLT is the magnetic local time coordinates (Shepherd, 2014), where the noon (12 MLT) is always fixed

20° N (pointed at by the white arrow). Such near-zero TEC values embedded in the dayside high TEC region (often called equatorial ionization anomaly) are physically unreasonable. A more reasonable imputation should keep the spatial continuity of TEC maps, especially in the dayside, high TEC value sub-regions. One can lower the ℓ_2-penalty in softImpute-ALS to improve the smoothness, thus increasing the rank, of the imputed map; but this results in over-fitting the observed region. We need the extra smoothness penalty to improve the missing region's smoothness while not overfitting the observed regions.

2. Large patches of unobserved values in the TEC map are poorly imputed by values near zero (see the red highlighted part in Figure 2.1(B).), even though the patches are clearly part of a sub-region with high TEC values.

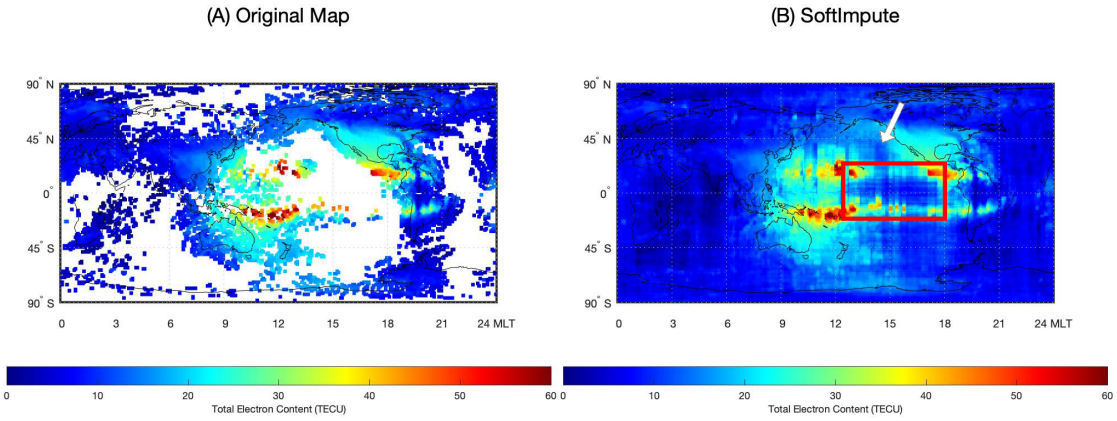


Figure 2.1: TEC maps: observed (left) and fitted by the SoftImpute approach (right).

The SoftImpute and a lot of other matrix completion methods, as we briefly mention a few below (especially work by Bell et al. (2007), Koren (2008) and Koren (2009)), are designed to deal with the imputation of matrices with scattered, random, non-patch missingness, such as building a recommendation system for `Netflix Prize` by Bennett and Lanning (2007). However, our data, the TEC maps, are scientific images and have non-random, auto-correlated, and patch missingness. Thus, these existing methods do not perform as desired in the TEC map completion problem.

Other related work include Mao and Saul (2004) and Lee and Seung (2000), which focused on non-negative matrix completion. Chen and Cichocki (2005) proposed non-negative matrix factorization with temporal smoothness, however, their method could not

at the center of each map while the locations of the continents and oceans are constantly shifting over time with the Earth's rotation.

deal with correlated spatial constraints. Barnes et al. (2009) proposed patch matching algorithm for image completion for nonparametric texture construction. However, the performance was not satisfactory when the original image lacked adequate data to complete the missing regions. Huang et al. (2014) extracted mid-level constraints and used them to guide the filling of missing regions. Yet the corrupted region must be small and relevant to visual data to have a good completion result. Our proposed method in this chapter, on the contrary, is capable of imputing a time series of matrices with a large number of missing values and guaranteeing both spatial smoothness and temporal consistency, which turns out to be very helpful for reconstructing scientific spatio-temporal data, e.g., the TEC maps/videos.

2.2 Methodology

In this section, we first summarize the two extensions we added on top of the softImpute-ALS method for our spatio-temporal tensor reconstruction task. We call the proposed method Video Imputation with SoftImpute, Temporal smoothing and Auxiliary data (VISTA). In Section 2.2.2, we present the details of the alternating minimization algorithm for estimation. Convergence properties of the algorithm are discussed in Section 2.2.3.

Following the notations we defined previously in Section 2.1, we consider a set of $m \times n$ matrices $\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_T\}$, each of which has missing values to be imputed. For each matrix $\mathbf{X}_t$, $1 \leq t \leq T$, the observed entries are labeled as 1 in the binary matrix $\mathbf{\Omega}_t$, and the missing entries are labeled as 1 in its complement $\mathbf{\Omega}_t^{\perp}$.

2.2.1 Extensions of the softImpute Method

To improve the softImpute-ALS matrix completion method to address the two issues described in Section 2.1.1, we propose a more general framework for matrix (video) completion, based on the softImpute-ALS algorithm and include it as a special case. In summary, our method is solving the following optimization problem:

$$\min_{\mathbf{A}_{1:T}, \mathbf{B}_{1:T}} \Bigg\{ F(\mathbf{A}_{1:T}, \mathbf{B}_{1:T}) \triangleq \frac{1}{2} \sum_{t=1}^{T} \|\mathrm{P}_{\mathbf{\Omega}_t}(\mathbf{X}_t - \mathbf{A}_t \mathbf{B}_t^{\top})\|_F^2 + \frac{\lambda_1}{2} \sum_{t=1}^{T} (\|\mathbf{A}_t\|_{\mathrm{F}}^2 + \|\mathbf{B}_t\|_{\mathrm{F}}^2) + \frac{\lambda_2}{2} \sum_{t=2}^{T} \|\mathbf{A}_t \mathbf{B}_t^{\top} - \mathbf{A}_{t-1} \mathbf{B}_{t-1}^{\top}\|_{\mathrm{F}}^2 + \frac{\lambda_3}{2} \sum_{t=1}^{T} \|\mathbf{Y}_t - \mathbf{A}_t \mathbf{B}_t^{\top}\|_{\mathrm{F}}^2 \Bigg\}, \tag{2.3}$$

where $\mathbf{Y}_1, \ldots, \mathbf{Y}_T$ are $m \times n$ auxiliary data with no missing values. Auxiliary data are obtained by applying certain smoothing algorithms over $\mathbf{X}_t$, which typically results in overly smoothed data thus not complying with the observations as desired. In our TEC map reconstruction problem, we smooth each TEC map $\mathbf{X}_t$ with a kernel smoothing method called Spherical Harmonics (SH), with relatively low orders of complexity, and the smoothed SH data are used as the auxiliary data. One can use the imputed maps out of an arbitrary imputation algorithm as the auxiliary data. We pick Spherical Harmonics for the spatial smoothness of its output. Backgrounds about Spherical Harmonics will be briefly introduced in section 2.3.1, and one can just think of the auxiliary data as a series of fully imputed maps for now. Note that our algorithm allows having this auxiliary data if such data exists and is justified within the scientific field. However, if such data is not available, we can simply set $\lambda_3 = 0$.

In (2.3), we add two additional regularization terms on top of (2.2), each serves as a solution to the two problems of softImpute-ALS. The term with λ_2 introduces temporal-smoothing (TS) to the imputation. This enables information sharing across neighboring time points. The term with λ_3 makes the imputation to learn from both the original observed data $\{\mathbf{X}_t\}$ and the auxiliary data $\{\mathbf{Y}_t\}$. Dropping both terms reduces the optimization problem to that in (2.2), which is solved via the original softImpute-ALS algorithm.

The reason for including the temporal-smoothing is that sub-regions with relatively high values tend to remain stable in adjacent frames, and penalizing the difference of imputed matrices between adjacent frames can eliminate the undesirable low-rank structure of the imputed sub-regions. The reason for learning from the auxiliary data is that the auxiliary data, such as the reconstructed map fitted with Spherical Harmonics, has reasonable "observations" in the large missing patches, thus providing additional information for imputing the large sub-regions with almost no observations. This is a special pattern of missingness that we face in the TEC map/video reconstruction task.

2.2.2 Description of Estimating Algorithm

In the previous section, we set up our matrix completion problem as an optimization problem (2.3). Although it is possible to solve (2.3) directly with off-the-shelf solvers, such solvers do not scale readily to large-scale problems. Following the approach in Hastie et al. (2015), we develop a majorization-minimization (MM) approach to solve (2.3) at scale. The main difference from Hastie et al. (2015) is that we have a sequence of matrices that we wish to complete jointly, so their algorithm must be modified, and its justification re-established. We give the details of the modifications in this section.

At a high level, the method is an optimized alternating least square (ALS) approach. The ALS method is commonly seen in matrix factorization (Paatero and Tapper, 1994; Kim and Park, 2008a,b) and matrix completion (Koren et al., 2009; Giménez-Febrer et al., 2019). Since we have more than two matrices, we update the factors $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_T, \mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_T$ one at a time following the order of: $\mathbf{A}_1 \to \mathbf{A}_2 \to \cdots \to \mathbf{A}_T \to \mathbf{B}_1 \to \mathbf{B}_2 \to \cdots \to \mathbf{B}_T \to \mathbf{A}_1 \to \mathbf{A}_2 \to \ldots$. This is an instance of cyclic block coordinate descent (Xu and Yin, 2013). In each step we update one factor, keeping the other $2T-1$ factors at their current values.

Suppose in the k-th iteration, we want to update $\mathbf{A}_t$. The current values for the other factors are: $\mathbf{A}_1^{(k+1)}, \mathbf{A}_2^{(k+1)}, \ldots, \mathbf{A}_{t-1}^{(k+1)}, \mathbf{A}_{t+1}^{(k)}, \ldots \mathbf{A}_T^{(k)}$ and $\mathbf{B}_1^{(k)}, \mathbf{B}_2^{(k)}, \ldots, \mathbf{B}_T^{(k)}$, respectively. Keeping every matrix other than $\mathbf{A}_t$ fixed at their current values, the convex optimization problem in (2.3) is reduced to the following minimization problem with respect to $\mathbf{A}_t$:

$$
\begin{aligned}
&\min_{\mathbf{A}_t} \Big\{ Q(\mathbf{A}_t | \mathbf{A}_{1:t-1}^{(k+1)}, \mathbf{A}_{t+1:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}) \\
&\triangleq \frac{1}{2}\|\mathrm{P}_{\mathbf{\Omega}_t}(\mathbf{X}_t - \mathbf{A}_t(\mathbf{B}_t^{(k)})^\top)\|_{\mathrm{F}}^2 + \frac{\lambda_1}{2}\|\mathbf{A}_t\|_{\mathrm{F}}^2 + \frac{\lambda_3}{2}\|\mathbf{Y}_t - \mathbf{A}_t(\mathbf{B}_t^{(k)})^\top\|_{\mathrm{F}}^2 \\
&\quad + \frac{\lambda_2}{2}\mathbf{I}_{\{t>1\}}\|\mathbf{A}_t(\mathbf{B}_t^{(k)})^\top - \mathbf{A}_{t-1}^{(k+1)}(\mathbf{B}_{t-1}^{(k)})^\top\|_{\mathrm{F}}^2 \\
&\quad + \frac{\lambda_2}{2}\mathbf{I}_{\{t<T\}}\|\mathbf{A}_{t+1}^{(k)}(\mathbf{B}_{t+1}^{(k)})^\top - \mathbf{A}_t(\mathbf{B}_t^{(k)})^\top\|_{\mathrm{F}}^2 \Big\},
\end{aligned} \tag{2.4}
$$

where $\mathbf{I}_{\{\cdot\}}$ is an indicator function that is equal to 1 if the condition in the subscript holds and zero otherwise[2]. Similarly for each $\mathbf{B}_t$, by keeping all factors other than $\mathbf{B}_t$ fixed, the optimization problem in (2.3) is reduced to the following minimization problem with respect to $\mathbf{B}_t$:

$$
\begin{aligned}
&\min_{\mathbf{B}_t} \Big\{ Q(\mathbf{B}_t | \mathbf{A}_{1:T}^{(k+1)}, \mathbf{B}_{1:t-1}^{(k+1)}, \mathbf{B}_{t+1:T}^{(k)}) \\
&\triangleq \frac{1}{2}\|\mathrm{P}_{\mathbf{\Omega}_t}(\mathbf{X}_t - \mathbf{A}_t^{(k+1)}\mathbf{B}_t^\top)\|_{\mathrm{F}}^2 + \frac{\lambda_1}{2}\|\mathbf{B}_t\|_{\mathrm{F}}^2 + \frac{\lambda_3}{2}\|\mathbf{Y}_t - \mathbf{A}_t^{(k+1)}\mathbf{B}_t^\top\|_{\mathrm{F}}^2 \\
&\quad + \frac{\lambda_2}{2}\mathbf{I}_{\{t>1\}}\|\mathbf{A}_t^{(k+1)}\mathbf{B}_t^\top - \mathbf{A}_{t-1}^{(k+1)}(\mathbf{B}_{t-1}^{(k+1)})^\top\|_{\mathrm{F}}^2 \\
&\quad + \frac{\lambda_2}{2}\mathbf{I}_{\{t<T\}}\|\mathbf{A}_{t+1}^{(k+1)}(\mathbf{B}_{t+1}^{(k)})^\top - \mathbf{A}_t^{(k+1)}\mathbf{B}_t^\top\|_{\mathrm{F}}^2 \Big\}.
\end{aligned} \tag{2.5}
$$

Both optimization problems in (2.4) and (2.5) are essentially ridge regression problems with multiple matrix responses. Finding solutions of (2.4) and (2.5), however, is not as

[2]We use $\mathbf{I}$ without the subscript to denote identity matrix in this chapter.

trivial as fitting a ridge regression since the projection operator $\mathrm{P}_{\mathbf{\Omega}_t}(\cdot)$ requires one to solve for $\mathbf{A}_t$ or $\mathbf{B}_t$ using the observed entries *only*.

To overcome this issue, we follow the idea in Hastie et al. (2015) and derive an upper bound for the first term in these optimization problems. For example, for (2.4), we have:

$$\begin{aligned}\|\mathrm{P}_{\mathbf{\Omega}_t}(\mathbf{X}_t - \mathbf{A}_t(\mathbf{B}_t^{(k)})^\top)\|_{\mathrm{F}}^2 &\leq \|\mathrm{P}_{\mathbf{\Omega}_t}(\mathbf{X}_t - \mathbf{A}_t(\mathbf{B}_t^{(k)})^\top) + \mathrm{P}_{\mathbf{\Omega}_t^\perp}(\mathbf{A}_t^{(k)}(\mathbf{B}_t^{(k)})^\top - \mathbf{A}_t(\mathbf{B}_t^{(k)})^\top)\|_{\mathrm{F}}^2 \\ &= \|\mathrm{P}_{\mathbf{\Omega}_t}(\mathbf{X}_t) + \mathrm{P}_{\mathbf{\Omega}_t^\perp}(\mathbf{A}_t^{(k)}(\mathbf{B}_t^{(k)})^\top) - \mathbf{A}_t(\mathbf{B}_t^{(k)})^\top\|_{\mathrm{F}}^2.\end{aligned} \tag{2.6}$$

Equality holds when $\mathbf{A}_t = \mathbf{A}_t^{(k)}$. This inequality is due to the fact that $\{(i,j)|[\mathbf{\Omega}_t]_{ij} = 1\} \cap \{(i,j)|[\mathbf{\Omega}_t^\perp]_{ij} = 1\} = \emptyset$, so the squared error in both $\{(i,j)|[\mathbf{\Omega}_t]_{ij} = 1\}$ and $\{(i,j)|[\mathbf{\Omega}_t^\perp]_{ij} = 1\}$ is at least as large as the squared error for $\{(i,j)|[\mathbf{\Omega}_t]_{ij} = 1\}$ only (the non-zero entries for the matrix on the left-hand-side is a subset of the non-zero entries for the matrix on the right-hand-side in the first line). We note that now the multi-response ridge regression problem can be solved directly since the unknown parameter $\mathbf{A}_t$ is not longer in the projection operator.

Define $\mathbf{X}_t^{(k)} = \mathrm{P}_{\mathbf{\Omega}_t}(\mathbf{X}_t) + \mathrm{P}_{\mathbf{\Omega}_t^\perp}(\mathbf{A}_t^{(k)}(\mathbf{B}_t^{(k)})^\top)$; i.e. $\mathbf{X}_t^{(k)}$ is an $m \times n$ matrix, with all observed entries keeping their values as those in $\mathbf{X}_t$ and all missing entries in $\mathbf{X}_t$ being filled in by $\mathbf{A}_t^{(k)}(\mathbf{B}_t^{(k)})^\top$. In light of (2.6), the objective function in (2.4) is upper bounded by:

$$\begin{aligned}&\widetilde{Q}(\mathbf{A}_t|\mathbf{A}_{1:t-1}^{(k+1)}, \mathbf{A}_{t:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}) \\ &\triangleq \frac{1}{2}\|\mathbf{X}_t^{(k)} - \mathbf{A}_t(\mathbf{B}_t^{(k)})^\top\|_{\mathrm{F}}^2 + \frac{\lambda_1}{2}\|\mathbf{A}_t\|_{\mathrm{F}}^2 + \frac{\lambda_3}{2}\|\mathbf{Y}_t - \mathbf{A}_t(\mathbf{B}_t^{(k)})^\top\|_{\mathrm{F}}^2 \\ &\quad + \frac{\lambda_2}{2}\mathbf{I}_{\{t>1\}}\|\mathbf{A}_t(\mathbf{B}_t^{(k)})^\top - \mathbf{A}_{t-1}^{(k+1)}(\mathbf{B}_{t-1}^{(k)})^\top\|_{\mathrm{F}}^2 \\ &\quad + \frac{\lambda_2}{2}\mathbf{I}_{\{t<T\}}\|\mathbf{A}_{t+1}^{(k)}(\mathbf{B}_{t+1}^{(k)})^\top - \mathbf{A}_t(\mathbf{B}_t^{(k)})^\top\|_{\mathrm{F}}^2.\end{aligned} \tag{2.7}$$

Similarly, define $\mathbf{X}_t^{(k+\frac{1}{2})} = \mathrm{P}_{\mathbf{\Omega}_t}(\mathbf{X}_t) + \mathrm{P}_{\mathbf{\Omega}_t^\perp}(\mathbf{A}_t^{(k+1)}(\mathbf{B}_t^{(k)})^\top)$. We have the following upper bound for the objective function in (2.5):

$$\begin{aligned}&\widetilde{Q}(\mathbf{B}_t|\mathbf{A}_{1:T}^{(k+1)}, \mathbf{B}_{1:t-1}^{(k+1)}, \mathbf{B}_{t:T}^{(k)}) \\ &\triangleq \frac{1}{2}\|\mathbf{X}_t^{(k+\frac{1}{2})} - \mathbf{A}_t^{(k+1)}\mathbf{B}_t^\top\|_{\mathrm{F}}^2 + \frac{\lambda_1}{2}\|\mathbf{B}_t\|_{\mathrm{F}}^2 + \frac{\lambda_3}{2}\|\mathbf{Y}_t - \mathbf{A}_t^{(k+1)}\mathbf{B}_t^\top\|_{\mathrm{F}}^2 \\ &\quad + \frac{\lambda_2}{2}\mathbf{I}_{\{t>1\}}\|\mathbf{A}_t^{(k+1)}\mathbf{B}_t^\top - \mathbf{A}_{t-1}^{(k+1)}(\mathbf{B}_{t-1}^{(k+1)})^\top\|_{\mathrm{F}}^2 \\ &\quad + \frac{\lambda_2}{2}\mathbf{I}_{\{t<T\}}\|\mathbf{A}_{t+1}^{(k+1)}(\mathbf{B}_{t+1}^{(k)})^\top - \mathbf{A}_t^{(k+1)}\mathbf{B}_t^\top\|_{\mathrm{F}}^2.\end{aligned} \tag{2.8}$$

We notice that these upper bounds have the property that

$$\widetilde{Q}(\mathbf{A}_t^{(k)} \mid \mathbf{A}_{1:t-1}^{(k+1)}, \mathbf{A}_{t:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}) = Q(\mathbf{A}_t^{(k)} \mid \mathbf{A}_{1:t-1}^{(k+1)}, \mathbf{A}_{t:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}),$$
$$\widetilde{Q}(\mathbf{B}_t^{(k)} \mid \mathbf{A}_{1:T}^{(k+1)}, \mathbf{B}_{1:t-1}^{(k+1)}, \mathbf{B}_{t:T}^{(k)}) = Q(\mathbf{B}_t^{(k)} \mid \mathbf{A}_{1:T}^{(k+1)}, \mathbf{B}_{1:t-1}^{(k+1)}, \mathbf{B}_{t:T}^{(k)}).$$

In other words, these upper bounds are tight at the current values of the $\mathbf{A}_t$ and $\mathbf{B}_t$. As we shall see, this property is crucial to the algorithm's convergence as it guarantees that minimizing the upper bound always reduces the objective value of (2.3).

The two upper bounds in (2.7) and (2.8) can be minimized to find the updated $\mathbf{A}_t, \mathbf{B}_t$:

$$\mathbf{A}_t^{(k+1)} = \arg\min \quad \left\{ \widetilde{Q}(\mathbf{A}_t | \mathbf{A}_{1:t-1}^{(k+1)}, \mathbf{A}_{t:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}) \right\}, \tag{2.9}$$
$$\mathbf{B}_t^{(k+1)} = \arg\min \quad \left\{ \widetilde{Q}(\mathbf{B}_t | \mathbf{A}_{1:T}^{(k+1)}, \mathbf{B}_{1:t-1}^{(k+1)}, \mathbf{B}_{t:T}^{(k)}) \right\}. \tag{2.10}$$

One may notice that in order to update the $\mathbf{A}_t, \mathbf{B}_t$, given the other matrices, we are essentially doing a multi-response ridge regression (Hoerl and Kennard, 1970) with shared coefficients and predictors. For example, for updating $\mathbf{A}_t$, with $\mathbf{B}_t^{(k)}$ as predictors, we are regressing on four responses: the "filled-in" matrix $\mathbf{X}_t^{(k)}$, the auxiliary data $\mathbf{Y}_t$, the imputation $\mathbf{A}_{t-1}^{(k+1)}(\mathbf{B}_{t-1}^{(k)})^\top$ at time $t-1$, and the imputation $\mathbf{A}_{t+1}^{(k)}(\mathbf{B}_{t+1}^{(k)})^\top$ at time $t+1$. The four responses are weighed by $1, \lambda_3, \lambda_2, \lambda_2$, respectively. Define a weighted label $\mathbf{Z}_t^{(k)}$ as:

$$\mathbf{Z}_t^{(k)} = \mathbf{X}_t^{(k)} + \lambda_2 \left(\mathbf{I}_{\{t>1\}} \mathbf{A}_{t-1}^{(k+1)} (\mathbf{B}_{t-1}^{(k)})^\top + \mathbf{I}_{\{t<T\}} \mathbf{A}_{t+1}^{(k)} (\mathbf{B}_{t+1}^{(k)})^\top \right) + \lambda_3 \mathbf{Y}_t. \tag{2.11}$$

Similarly, when we update $\mathbf{B}_t$, we define $\mathbf{Z}_t^{(k+\frac{1}{2})}$ as:

$$\mathbf{Z}_t^{(k+\frac{1}{2})} = \mathbf{X}_t^{(k+\frac{1}{2})} + \lambda_2 \left(\mathbf{I}_{\{t>1\}} \mathbf{A}_{t-1}^{(k+1)} (\mathbf{B}_{t-1}^{(k+1)})^\top + \mathbf{I}_{\{t<T\}} \mathbf{A}_{t+1}^{(k+1)} (\mathbf{B}_{t+1}^{(k)})^\top \right) + \lambda_3 \mathbf{Y}_t. \tag{2.12}$$

With the notations above, the updated $\mathbf{A}_t$ in (2.9) has a closed-form solution:

$$\mathbf{A}_t^{(k+1)} = \left[(1 + \lambda_2(\mathbf{I}_{\{t<T\}} + \mathbf{I}_{\{t>1\}}) + \lambda_3)(\mathbf{B}_t^{(k)})^\top \mathbf{B}_t^{(k)} + \lambda_1 \mathbf{I} \right]^{-1} \mathbf{Z}_t^{(k)} \mathbf{B}_t^{(k)}. \tag{2.13}$$

Similarly, the updated $\mathbf{B}_t$ in (2.10) also has a closed-form solution:

$$\mathbf{B}_t^{(k+1)} = \left[(1 + \lambda_2(\mathbf{I}_{\{t<T\}} + \mathbf{I}_{\{t>1\}}) + \lambda_3)(\mathbf{A}_t^{(k+1)})^\top \mathbf{A}_t^{(k+1)} + \lambda_1 \mathbf{I} \right]^{-1} (\mathbf{Z}_t^{(k+\frac{1}{2})})^\top \mathbf{A}_t^{(k+1)}. \tag{2.14}$$

The two closed-form solutions for $\mathbf{A}_t^{(k+1)}, \mathbf{B}_t^{(k+1)}$ are shrinkage estimators (Efron et al. (1976); Lehmann and Casella (2006)). Compared to the original softImpute-ALS method, the shrinkage towards zero (which is dictated by the regularization term for $\mathbf{A}$ and $\mathbf{B}$ ma-

trices) is not as large if given the same λ_1 and non-zero λ_2, λ_3 values. Therefore, one can expect imputation matrices with higher rank out of the algorithm with either temporal smoothing ($\lambda_2 > 0$) or auxiliary data ($\lambda_3 > 0$). In other words, the presence of any non-zero λ_2 and λ_3 make the shrinkage effect introduced by λ_1 weaker. One can easily lower the value of λ_1 to make the shrinkage of $\mathbf{A}$ and $\mathbf{B}$ weaker without introducing λ_2 or λ_3. However, the softImpute-ALS algorithm itself (when $\lambda_2 = \lambda_3 = 0$) cannot guarantee temporal and spatial smoothness and may over-fit the observed entries. This shrinkage formula exemplifies the role that each component, including the ℓ_2-regularization, the auxiliary data, and the temporal smoothing, plays. The roles of the tuning parameters $\lambda_1, \lambda_2, \lambda_3$ are also clear from this shrinkage formula: the relative values of these three tuning parameters determine the extent of the sparsity of the imputed matrices, the extent of the temporal smoothing, and the weight of the auxiliary data.

The two shrinkage formulae for $\mathbf{A}_t$ and $\mathbf{B}_t$ showcase a mutual normalization phenomenon: the extent of the shrinkage in estimating $\mathbf{A}_t$ depends on the current values of $\mathbf{B}_t$ and vice versa. This results from the matrix factorization assumption and is quite interesting from the perspective of empirical Bayes estimators. One can easily formulate the optimization problem in (2.3) as solving the maximum-a-posterior (MAP) estimate of a Bayesian model with latent factored Markovian structures. In that case, λ_1^{-1}, λ_2^{-1}, and λ_3^{-1} serve as the prior variance for $\mathbf{A}_t, \mathbf{B}_t$, the variance of the Gaussian Markovian latent structure, and the variance of the auxiliary data. We omit further details of this here to not deviate too much from the main results of the current chapter.

Additionally, compared with existing matrix completion methods, we use weighted labels $\mathbf{Z}_t^{(k)}, \mathbf{Z}_t^{(k+\frac{1}{2})}$ that include imputations of neighboring time points on top of the original data when learning the imputation. A related work (Wang et al., 2014) also accounts for temporal smoothness in matrix completion, but they use alternating direction method of multipliers (ADMM) (Chen et al., 2012). Also, our weighted label incorporates the auxiliary data that are particularly helpful for imputing scientific images with large patch missingness.

In a single iteration, we update $\mathbf{A}_t$ using (2.13) for each t and update $\mathbf{B}_t$ using (2.14) for each t. The algorithm terminates when all imputations $\mathbf{A}_t^{(k)}(\mathbf{B}_t^{(k)})^\top$ converge. To verify convergence, we calculate the relative change of the Frobenius norm:

$$\nabla F_t^{(k)} = \frac{\|\mathbf{A}_t^{(k+1)}(\mathbf{B}_t^{(k+1)})^\top - \mathbf{A}_t^{(k)}(\mathbf{B}_t^{(k)})^\top\|_{\mathrm{F}}^2}{\|\mathbf{A}_t^{(k)}(\mathbf{B}_t^{(k)})^\top\|_{\mathrm{F}}^2}. \tag{2.15}$$

The termination rule that we set for the iterative algorithm is that the algorithm stops at iteration k if $\max\{\nabla F_1^{(k)}, \nabla F_2^{(k)}, \ldots, \nabla F_T^{(k)}\}$ is smaller than a pre-specified threshold.

The full algorithm uses a singular value decomposition (SVD) form to express $\mathbf{A}_t, \mathbf{B}_t$: $\mathbf{A}_t = \mathbf{U}_t\mathbf{D}_t, \mathbf{B}_t = \mathbf{V}_t\mathbf{D}_t$, following the original softImpute algorithm. This guarantees $\|\mathbf{A}_t\mathbf{B}_t^\top\|_* = \frac{1}{2}(\|\mathbf{A}_t\|_{\mathrm{F}}^2 + \|\mathbf{B}_t\|_{\mathrm{F}}^2)$, which means that the algorithm is equally penalizing the trace norm of the imputation. We present the full algorithm below in Algorithm 2.1. The last few steps in the algorithm adopt the idea of softImpute-ALS in Hastie et al. (2015) to ensure that the output imputation matrix is sparse and has zero singular values.

Algorithm 2.1 softImpute-ALS with Temporal Smoothing and Auxiliary Data

Input: $m \times n$ Sparse data $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_T$, $m \times n$ auxiliary data $\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_T$, operating rank r. Maximum iteration K and convergence threshold τ.

Initialization: For $1 \le t \le T$, $\mathbf{A}_t^{(1)} = \mathbf{U}_t\mathbf{D}_t, \mathbf{B}_t^{(1)} = \mathbf{V}_t\mathbf{D}_t$, where $\mathbf{U}_t, \mathbf{V}_t$ are $m \times r, n \times r$ random matrices with orthogonal columns. $\mathbf{D}_t$ is an $r \times r$ identity matrix.

Update A:

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

a. Let $\mathbf{X}_t^{(k)} = \mathrm{P}_{\mathbf{\Omega}_t}(\mathbf{X}_t) + \mathrm{P}_{\mathbf{\Omega}_t^\perp}(\mathbf{A}_t^{(k)}(\mathbf{B}_t^{(k)})^\top)$, which is the "filled-in" version of $\mathbf{X}_t$.

b. Let $\mathbf{Z}_t^{(k)}$ be the weighted label in equation (2.11).

c. $\mathbf{A}_t^{(k+1)}$ is updated as equation (2.13).

end for

Update B: For every t, repeat a,b,c steps above, with $\mathbf{X}_t^{(k)}, \mathbf{Z}_t^{(k)}$ being replace by $\mathbf{X}_t^{(k+\frac{1}{2})}, \mathbf{Z}_t^{(k+\frac{1}{2})}$. $\mathbf{B}_t^{(k+1)}$ is calculated following equation (2.14).

Repeat updating $\mathbf{A}_{1:T}$ and $\mathbf{B}_{1:T}$ until convergence. The algorithm converges when $\max\{\nabla F_1^{(k)}, \nabla F_2^{(k)}, \ldots, \nabla F_T^{(k)}\} < \tau$, with $\nabla F_t^{(k)}$ defined in (2.15).

For any t, denote the final output as $\mathbf{A}_t^*, \mathbf{B}_t^*$. Let $\mathbf{X}_t^* = \mathrm{P}_{\mathbf{\Omega}_t}(\mathbf{X}_t) + \mathrm{P}_{\mathbf{\Omega}_t^\perp}(\mathbf{A}_t^*(\mathbf{B}_t^*)^\top)$.

Do SVD for $\mathbf{A}_t^*(\mathbf{B}_t^*)^\top = \mathbf{U}_t^*(\mathbf{D}_t^*)^2(\mathbf{V}_t^*)^\top$.

Define $\mathbf{M}_t = \mathbf{X}_t^*\mathbf{V}_t^*$ and do SVD for $\mathbf{M}_t = \widetilde{\mathbf{U}}_t\widetilde{\mathbf{D}}_t\mathbf{R}_t^\top$.

Do soft-thresholding on $\widetilde{\mathbf{D}}_t$: $\widetilde{\mathbf{D}}_{t,\lambda_1} = \mathbf{diag}\,[(\sigma_1 - \lambda_1)_+, (\sigma_2 - \lambda_1)_+, \ldots, (\sigma_r - \lambda_1)_+]$.

Output imputation for time t as $\widehat{\mathbf{X}}_t = \widetilde{\mathbf{U}}_t\widetilde{\mathbf{D}}_{t,\lambda_1}(\mathbf{V}_t^*\mathbf{R}_t)^\top$.

Output: Imputed matrices $\widehat{\mathbf{X}}_1, \ldots, \widehat{\mathbf{X}}_T$.

Before moving on to the discussion of the theoretical property of the algorithm, we want to note that there are three hyper-parameters in our algorithm: $\lambda_1, \lambda_2, \lambda_3$, and potentially more hyper-parameters in the process of generating the auxiliary data. It would be computationally cumbersome if one wants to search for the best parameters via grid search and cross-validation. Based on our empirical experience, our method provides stable imputation results for a wide range of hyper-parameters. Therefore, we recommend our readers to do a sequential search: search for the best λ_1 first, and then search for the best λ_2 and λ_3 in parallel. One can determine the initial range of these parameters using a relatively small dataset (i.e. with only a few frames) and it typically generalizes well to the entire dataset. In the following texts, we use this sequential search method and pick the best hyper-

parameters based on the performance on a held-out validation set. Specifically for the TEC map reconstruction problem, we recommend $\lambda_1 \in [0.8, 1.0], \lambda_2 \in [0.2, 0.3], \lambda_3 \in [0.02, 0.03]$.

Another tuning parameter is r, which is the other dimension of the factor matrices $\mathbf{A}$ and $\mathbf{B}$ and is set to $\min\{m, n\}$ in our algorithm. By doing so, we allow for the maximum possible rank for the imputed map. In other applications, if one has prior knowledge about the rank of the data, one can set an initial low-rank structure with $r < \min\{m, n\}$. Even though we do not take advantage of this in the TEC imputation task, we still make it a modeling possibility for other applications.

2.2.3 Theoretical Properties of the Algorithm

In this section, we provide theoretical results for the convergence rate of the algorithm and show that the algorithm converges to a stationary point of the problem defined in equation (2.3). Proofs, following the techniques in Hastie et al. (2015), are included in Appendix A.

Recall that the objective function to minimize is defined in (2.3), which is $F(\mathbf{A}_{1:T}, \mathbf{B}_{1:T})$. We approach this optimization problem by the majorization-minimization algorithm with alternating least squares (ALS). The first theoretical result, presented in Theorem 2.2.1, indicates that each update on $\mathbf{A}_t$ and $\mathbf{B}_t$ does not increase the objective function value.

Let $\{\mathbf{A}_{1:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}\}_{k\geq 1}$ be the sequence of $\mathbf{A}_{1:T}, \mathbf{B}_{1:T}$ generated through the iterations of Algorithm 2.1, where $\mathbf{A}_{1:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}$ denotes the matrices at iteration k. Define the descent of objective function value at iteration k as $\Delta_k = F(\mathbf{A}_{1:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}) - F(\mathbf{A}_{1:T}^{(k+1)}, \mathbf{B}_{1:T}^{(k+1)})$.

Theorem 2.2.1. *The value of the objective function is non-increasing, i.e.,*

$$F(\mathbf{A}_{1:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}) \geq F(\mathbf{A}_{1:T}^{(k+1)}, \mathbf{B}_{1:T}^{(k)}) \geq F(\mathbf{A}_{1:T}^{(k+1)}, \mathbf{B}_{1:T}^{(k+1)}),$$

thus $\Delta_k \geq 0$, *for all* $k \geq 1$.

Proof of this theorem is in Section A.1 of the Appendix.

Theorem 2.2.1 indicates that each iteration is making the objective function smaller. Given that each matrix $\mathbf{A}_t, \mathbf{B}_t$ is updated by ridge regression, we further show that, in Theorem 2.2.2, the descent of the objective function value in each iteration has a lower bound.

Theorem 2.2.2. *We have the following lower bound for Δ_k:*

$$\Delta_k \geq \frac{\lambda_1}{2}\sum_{t=1}^{T}\left(\|\mathbf{A}_t^{(k)}-\mathbf{A}_t^{(k+1)}\|_{\mathrm{F}}^2+\|\mathbf{B}_t^{(k)}-\mathbf{B}_t^{(k+1)}\|_{\mathrm{F}}^2\right) + \frac{1}{2}\sum_{t=1}^{T}(1+\lambda_2(1+\mathbf{I}_{\{2\leq t\leq T-1\}})+\lambda_3)\left[\delta_{k,t}\right], \tag{2.16}$$

where $\delta_{k,t}=\|(\mathbf{A}_t^{(k)}-\mathbf{A}_t^{(k+1)})(\mathbf{B}_t^{(k)})^\top\|_{\mathrm{F}}^2+\|\mathbf{A}_t^{(k+1)}(\mathbf{B}_t^{(k)}-\mathbf{B}_t^{(k+1)})^\top\|_{\mathrm{F}}^2$.

The proof of this theorem is in Section A.2 of the Appendix.

This result gives a lower bound for the descent of the objective function value at iteration step k. In the first term, with a non-zero λ_1, as long as there exists one $\mathbf{A}_t$ or $\mathbf{B}_t$ that has a different value/entry before and after the update, the lower bound is greater than zero. Thus Δ_k may be interpreted as a measure of the optimality of $\{(\mathbf{A}_t^{(k)},\mathbf{B}_t^{(k)})\}_{t=1}^T$. Given the result in theorem 2.2.1 and 2.2.2, the sequence of objective function values is a bounded, strictly monotonic sequence, thus having a finite limit (may not be unique, depending on the initialization), denoted as f^∞. The following result, Theorem 2.2.3, gives the convergence rate for our algorithm.

Theorem 2.2.3. *Let the limit of the objective function $F(\mathbf{A}_{1:T}^{(k)},\mathbf{B}_{1:T}^{(k)})$ be f^∞, we have:*

$$\min_{1\leq k\leq K}\Delta_k \leq \frac{F(\mathbf{A}_{1:T}^{(1)},\mathbf{B}_{1:T}^{(1)})-f^\infty}{K}, \tag{2.17}$$

where K is the total number of iterations. Additionally, assume that there exists positive constants l^L and l^U such that $l^L\mathbf{I}\leq(\mathbf{A}_t^{(k)})^\top\mathbf{A}_t^{(k)}\leq l^U\mathbf{I}$, $l^L\mathbf{I}\leq(\mathbf{B}_t^{(k)})^\top\mathbf{B}_t^{(k)}\leq l^U\mathbf{I}$ for all t,k, then we have:

$$\min_{1\leq k\leq K}\left\{\sum_{t=1}^{T}\left(\|\mathbf{A}_t^{(k)}-\mathbf{A}_t^{(k+1)}\|_{\mathrm{F}}^2+\|\mathbf{B}_t^{(k)}-\mathbf{B}_t^{(k+1)}\|_{\mathrm{F}}^2\right)\right\} \leq \frac{2}{(1+\lambda_2+\lambda_3)l^L+\lambda_1}\left(\frac{F(\mathbf{A}_{1:T}^{(1)},\mathbf{B}_{1:T}^{(1)})-f^\infty}{K}\right), \tag{2.18}$$

and

$$\min_{1\leq k\leq K}\left\{\sum_{t=1}^{T}\left(\|(\mathbf{A}_t^{(k)}-\mathbf{A}_t^{(k+1)})(\mathbf{B}_t^{(k)})^\top\|_{\mathrm{F}}^2+\|\mathbf{A}_t^{(k+1)}(\mathbf{B}_t^{(k)}-\mathbf{B}_t^{(k+1)})^\top\|_{\mathrm{F}}^2\right)\right\} \leq \frac{2l^U}{l^U(1+\lambda_2+\lambda_3)+\lambda_1}\left(\frac{F(\mathbf{A}_{1:T}^{(1)},\mathbf{B}_{1:T}^{(1)})-f^\infty}{K}\right). \tag{2.19}$$

The proofs are very similar to that of Theorem 4 and Corollary 1 of Hastie et al. (2015). Thus we only briefly describe the proof in Section A.3 of the Appendix.

Recall Δ_k is a measure of the optimality of $\{(\mathbf{A}_t^{(k)}, \mathbf{B}_t^{(k)})\}_{t=1}^T$. Theorem 2.2.3 implies this measure of optimality converges at a rate of $O(1/K)$. We note that this is the same rate of convergence as softImpute-ALS, which is $O(1/K)$. There are two additional notions of convergence, namely (2.18) and (2.19), that show the role of $\lambda_1, \lambda_2, \lambda_3$ in the convergence rate. Generally, fewer iterations are needed given larger $\lambda_1, \lambda_2, \lambda_3$.

The last result follows the Theorem 5 in Hastie et al. (2015), which states that the limit point of the sequence $\{\mathbf{A}_{1:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}\}_{k\geq 1}$, denoted by $\mathbf{A}_{1:T}^*, \mathbf{B}_{1:T}^*$, is a stationary point of problem (2.3). We state the result in Theorem 2.2.4 here without a proof since the technique is the same as Hastie et al. (2015).

Theorem 2.2.4. *Let $\{\mathbf{A}_{1:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}\}_{k\geq 1}$ be the sequence of $\mathbf{A}_{1:T}, \mathbf{B}_{1:T}$ generated throughout the iterations of Algorithm 2.1. For $\lambda_1 > 0$, the limit point of the sequence $\mathbf{A}_{1:T}^*, \mathbf{B}_{1:T}^*$ is a stationary point of problem (2.3) in the sense that:*

$$\mathbf{A}_t^* = \arg\min_{\mathbf{A}_t} \left\{ \widetilde{Q}(\mathbf{A}_t | \mathbf{A}_{1:T}^*, \mathbf{B}_{1:T}^*) \right\},$$
$$\mathbf{B}_t^* = \arg\min_{\mathbf{B}_t} \left\{ \widetilde{Q}(\mathbf{B}_t | \mathbf{A}_{1:T}^*, \mathbf{B}_{1:T}^*) \right\}.$$

Just as the softImpute paper (Hastie et al., 2015), there is no theoretical guarantee that the sequence of matrices generated by alternating least squares converges to the global minimum of the optimization problem (2.3). What we have proved and stated in this section is that the algorithm improves the objective function at each iteration, converges to a stationary point, and the three tuning parameters have their specific roles in the rate of convergence. In practice, our algorithm performs well on the video/matrix imputation task, as we will show later.

Since we use the cyclic least square method on updating $2T$ matrices, the computational cost of the algorithm is about T times the cost of softImpute-ALS which updates 2 matrices per iteration. But our approach imputes T matrices simultaneously, the computational cost is thus similar to softImpute-ALS per iteration after being scaled by the number of parameters. In practice, the algorithm converges in fewer iterations when including the temporal smoothing and auxiliary data penalty, so the algorithm is indeed less time-consuming than softImpute-ALS.

In the next section, we use a carefully designed simulation study to compare our method with the baseline softImpute-ALS in terms of the accuracy of the imputation. Then we present the imputation results for TEC maps using our method and demonstrate how

our method resolves the two issues with softImpute-ALS when imputing TEC videos described in Section 2.1.1.

2.3 Numerical Studies

To compare our proposed method with the softImpute-ALS and the state-of-the-art method in the scientific field for TEC completion (i.e. the Spherical Harmonics method), we use the TEC maps provided by the International GNSS Service (IGS) as the data based on which we design a simulation experiment. The advantage of using this data is that there are no missing values. Therefore, we can create missing values artificially and still know the original values of the missing entries. We pick the 1-day IGS data on Sept-08, 2017, which contains 96 matrices at a 15-min resolution. Each TEC map is of size 71×73, so the video is of size $71 \times 73 \times 96$. We resize each TEC map to 181×361 with bi-linear interpolation to match the size of the TEC map of the Madrigal database, which is the observed TEC map with missing values that we want to impute eventually.

In this section, we describe 4 different ways of "creating" missing values in the IGS data. After creating missingness, we then generate auxiliary data and apply our imputation algorithms. In Figure 2.2, we illustrate the data pipeline from any input video (with missing values) to the output video (imputed full videos). This pipeline is applied to all numerical analyses and empirical analyses in the chapter.

We apply the Box-Cox transformation (Box and Cox, 1964) on each observed pixel of the input video and auxiliary data (i.e. spherical harmonics fitted data) to make the data more normally distributed. Pixel-wisely, the Box-Cox transformation is doing $y' = (y^\lambda - 1)/\lambda$ for any pixel intensity y. This could make the imputation more robust to extreme values.

Before going into details of the design and results of numerical analysis, we first give a brief overview of the spherical harmonics fitting method, which is the method we use to generate auxiliary data from the input video.

2.3.1 A Brief Note on Spherical Harmonics Fitting

Known as the angular portion of the solutions to the Laplace's equation in spherical coordinates, spherical harmonic functions define a complete orthogonal basis for functions on a unit sphere and therefore can approximate a sufficiently smooth surface function. The spherical harmonics is approximating a function in the spherical coordinate system

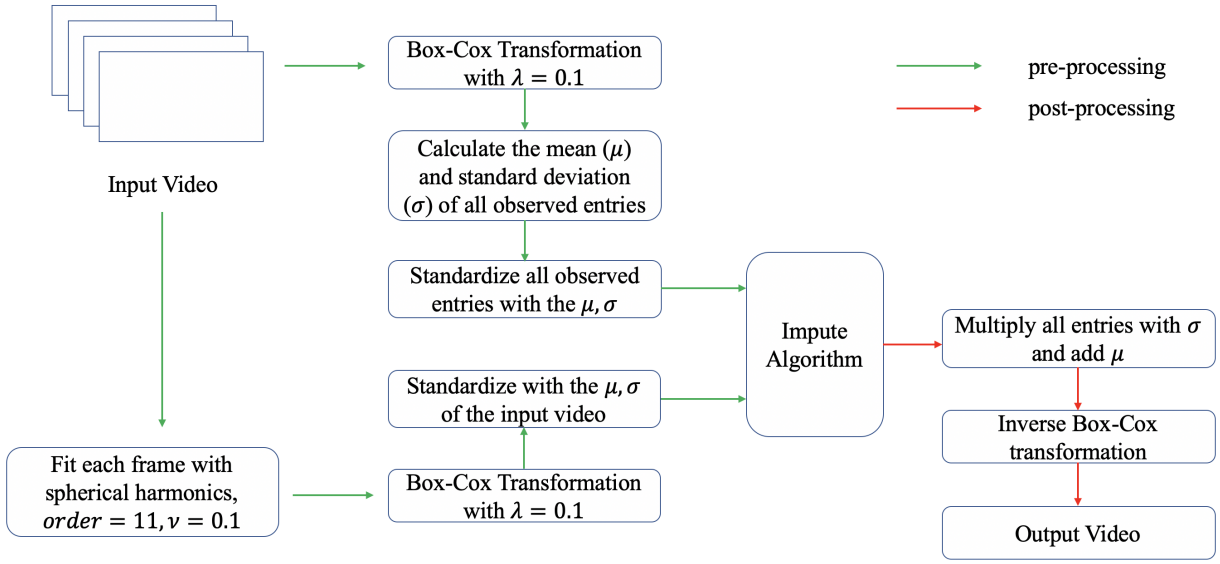


Figure 2.2: Data analysis pipeline: video imputation. The input video contains missing values. Spherical Harmonics is fitted on the input video with a carefully chosen order $l_{\max}$ and ℓ_2 regularization weight ν to optimize its performance. Standardization is done for all observed pixels in both data. To obtain the output video, we inverted both standardization and the Box-Cox transformation to make sure the input and output videos have comparable scales.

$f(\theta, \phi)$ via a basis expansion of the form:

$$f(\theta, \phi) \approx \sum_{l=0}^{l_{max}} \sum_{m=-l}^{l} a_l^m Y_l^m(\theta, \phi),$$

where θ and ϕ are the elevation and azimuth angles in the spherical coordinates. $Y_l^m(\theta, \phi)$ denotes a spherical harmonic function with degree m and order l ($|m| \leq l$), and a_l^m is the corresponding coefficient. Similar to the Fourier series, when the maximum order $l_{max} \to \infty$, the expansion becomes an exact representation of the function $f(\theta, \phi)$.

By viewing the global TEC distribution at a given time as a function of latitude and longitude, we can use the spherical harmonic expansion as an approximation to the complete TEC map. Every single measurement of TEC at location (θ_i, ϕ_i) can provide a linear equation $f(\theta_i, \phi_i) = \text{TEC}_i = \sum_{l=0}^{l_{\max}} \sum_{m=-l}^{l} a_l^m Y_l^m(\theta_i, \phi_i)$, and by solving a system of linear equations based on all the available measurements on a TEC map, a set of coefficients a_l^m can be obtained for a given l_{max}, and the resulting expansion is the least squares approximation of the global TEC map (see Figure 2.3 for a concrete example). To avoid high-frequency artifacts and negative TEC values in the spherical harmonic fitting result, an ℓ_2-penalty over all $a_l^m, l \leq l_{\max}, |m| \leq l$ and an inequality constraint over the resulting

approximation (Zhang et al., 2013) were applied when solving the least squares problem.

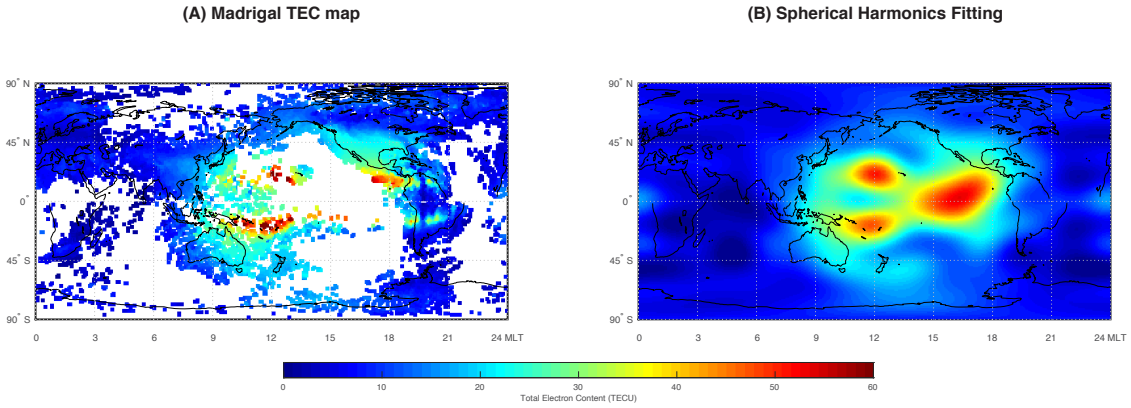


Figure 2.3: (A) Madrigal TEC map with missing data and (B) complete TEC map approximated by the spherical harmonics expansion.

As one can see from Figure 2.3, the fitted map provides some reasonable "observations" in the large missing patches, which could potentially improve the imputation in the oceanic areas when using the softImpute approach. In our analysis, we fit any TEC map with missing values using spherical harmonics to get the auxiliary data for the TEC map, and we choose the order $l_{max} = 11$ and the tuning parameter of the ℓ_2-regularization as $\nu = 0.1$ based on cross-validation. Chapter 4 and the corresponding Appendix C.6 contain additional details about spherical harmonics and the basis functions.

2.3.2 Description of the Design of Numerical Experiments

Given the resized $181 \times 361 \times 96$ TEC tensor from the IGS dataset, we introduce four data missingness patterns in each of the 96 matrices for numerical analysis. An example of the original TEC map with various artificial missingness patterns is shown in Figure 2.4.

1. Random missingness (sub-figure B): for each matrix, randomly drop $30\%/50\%/70\%$ of the pixels.

2. Temporal missingness (sub-figure C): for $t = 1$, randomly drop $30\%/50\%/70\%$ of pixels and let the missing mask move 6 columns horizontally (direction shown as the red arrow) per frame.

3. Random patch missingness (sub-figure E): for each frame, randomly pick a center on a fixed bounding box around a high TEC value region (the red bounding box in sub-figure D) and create a 27×27 or 45×45 or 63×63 patch as missing.

4. Temporal patch missingness (sub-figure F): similar to patch missingness, but the center of the $27 \times 27/45 \times 45/63 \times 63$ patch moves along the bounding box at the speed of 6 columns or rows per frame (anti-clockwise as shown by the red arrow).

Random/temporal missingness is designed to emulate a random/auto-correlated, scattered data missingness pattern as can be observed in some local regions of the TEC map. Random/temporal patch missingness simulates the large patch missingness in the TEC map. The difference is that temporal patch missingness has largely overlapping patches in temporally adjacent matrices, further restricting the information available from the neighboring time points. In the Madrigal database, the missing pattern is a mixture of scattered-missing and patch-missing, and mainly patch-missing. In the original context where softImpute is applied, such as the Netflix competition, scattered missingness is more common.

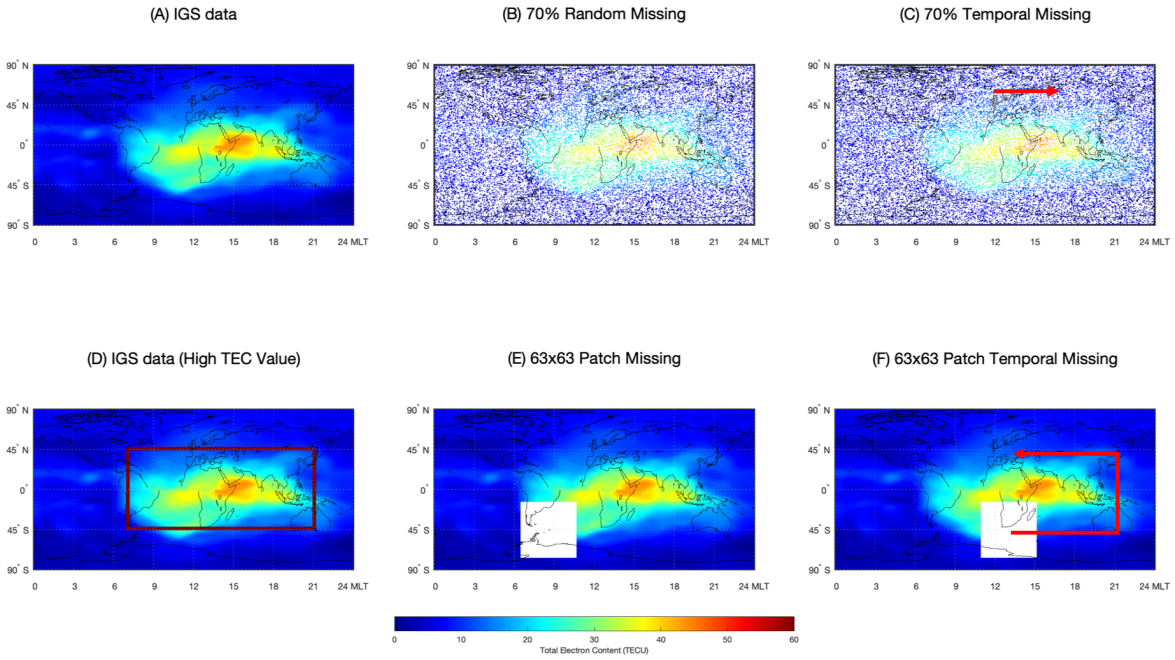


Figure 2.4: Four missingness patterns, where white pixels denote missing values. (A) 181×361 TEC map (IGS data) at 2017-09-08 11:57:30 UT. (B) Pattern 1: Random missingness. (C) Pattern 2: Temporal missingness. (D) 181×361 TEC map, with a bounding box around region $[45°\text{N}, 45°\text{S}] \times [7\ \text{MLT}, 21\ \text{MLT}]$ with high TEC values. (E) Pattern 3: Random Patch missingness. (F) Pattern 4: Temporal patch missingness.

2.3.3 Results from Numerical Studies

After applying each data missingness pattern to the IGS data, we can run softImpute-ALS and our proposed algorithm VISTA with various choices of hyper-parameters to im-

pute the data. To evaluate model performance on imputing the missing values, we use the relative squared error (RSE) (as used in Liu et al. (2012); Wang et al. (2014)):

$$\mathrm{RSE}(\mathbf{X}_t, \mathbf{X}_t^*, \mathbf{\Omega}_t) = \frac{\|\mathrm{P}_{\mathbf{\Omega}_t^\perp}(\mathbf{X}_t^* - \mathbf{X}_t)\|_\mathrm{F}}{\|\mathrm{P}_{\mathbf{\Omega}_t^\perp}(\mathbf{X}_t)\|_\mathrm{F}},$$

where $\mathbf{X}_t$ is the fully-observed IGS data, $\mathbf{\Omega}_t$, $\mathrm{P}_{\mathbf{\Omega}_t^\perp}(.)$ follow the definition in section 2.1, $\mathbf{X}_t^*$ is the imputation of $\mathbf{X}_t$ with missing values. The RSE measures the imputation performance on the missing pixels. Throughout the rest of this chapter, we report the RSE in the unit of percentages (%).

For each data missingness pattern and each level of missingness, we fit four models to impute the matrices, where each model is a variant of the optimization problem in equation (2.3). Here is a list of the four models that we compare and their acronyms.

1. **soft**: softImpute as in Hastie et al. (2015): $\lambda_1 = 0.9, \lambda_2 = 0, \lambda_3 = 0$.
2. **TS**: softImpute + temporal smoothing: $\lambda_1 = 0.9, \lambda_2 = 0.05, \lambda_3 = 0$.
3. **SH**: softImpute + auxiliary data based on spherical harmonics: $\lambda_1 = 0.9, \lambda_2 = 0, \lambda_3 = 0.01$.
4. **TS+SH**: softImpute + temporal smoothing + auxiliary data based on spherical harmonics: $\lambda_1 = 0.9, \lambda_2 = 0.05, \lambda_3 = 0.01$.

The tuning parameters above are chosen for demonstration purposes instead of being selected from an extensive grid search as we do for the real data. The reason why we set all $\lambda_1 = 0.9$ is because as one will see in the empirical section, this value works well for the baseline softImpute method when imputing the TEC maps in general. Therefore, our choice of the tuning parameters optimizes the performance of the softImpute algorithm. We will show that even in this setting, our newly proposed algorithm outperforms in the majority of cases. And we fix $\lambda_2 = 0.05$ and $\lambda_3 = 0.01$, which are not the parameters that give the best performances. We pick these parameters just to show that the method can produce decent results even when one does not fine-tune the hyper-parameters.

For each of the 96 matrices of the IGS data, we calculate the RSE on the missing pixels. Suppose that the RSE of the imputed matrix at time t for the four models are $\mathrm{RSE}_t^{(\mathrm{soft})}, \mathrm{RSE}_t^{(\mathrm{TS})}, \mathrm{RSE}_t^{(\mathrm{SH})}, \mathrm{RSE}_t^{(\mathrm{TS+SH})}$, respectively. We define the test set RSE margin over softImpute for the three variants of our model as $\Delta\mathrm{RSE}_t^{(k)} = \mathrm{RSE}_t^{(\mathrm{soft})} - \mathrm{RSE}_t^{(k)}, k \in \{\mathrm{TS,SH,TS+SH}\}$. Then $\Delta\mathrm{RSE}_t^{(k)} > 0$ means that model k performs, on average, better than the softImpute method on imputing the missing value of matrix at time t. In Figure 2.5,

we report the average margin over softImpute across all 96 matrices for three variants of our model under the random missing and temporal missing scenarios.

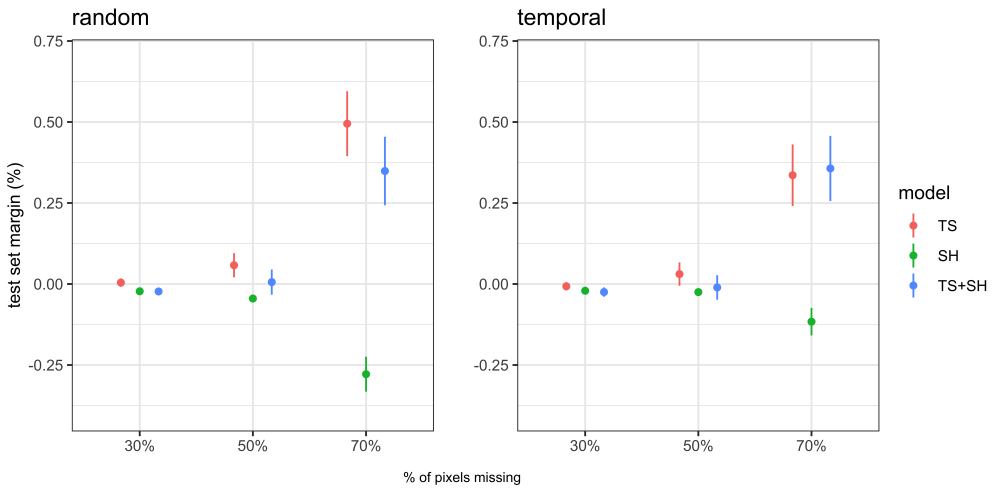


Figure 2.5: Numerical Analysis: random missing and temporal missing results. Three variants of our method are considered: TS, SH, and TS+SH. A detailed explanation is included in the main text. The scatter points show the average test set RSE margin over the baseline softImpute method, positive means performance better than softImpute. Error bar gives the 95% confidence interval.

As one can see, as the level of missingness becomes higher, the temporal smoothing (TS) model and the full model (TS+SH) start to build a positive margin while the spherical harmonics (SH) model performs worse than the softImpute. However, the corresponding margins, regardless of being positive/negative, are all close to zero. This is mostly because, under scattered missingness, the dominant information source for imputing missing entries is from neighbors that are spatially close thus temporal smoothing cannot help much. Spherical harmonics, on the other hand, has over-smoothed the data and gives some misleading information on the missing entries but the overfitting is not very severe.

The more interesting scenario arises when we introduce the patch missingness, which resembles the real data more. In Figure 2.6, we make a similar plot as Figure 2.5. We observe drastic differences between the scales of the two plots. With patch missingness, all three models perform significantly better than softImpute, with the smallest margin being around 4%. Similarly, the higher the level of missingness, the greater the margin.

What's even more intriguing is the difference between random patch missing and temporal patch missing. When patches are randomly missing, the missing patch in the matrix at t is very likely to be fully or partially observed in the matrices at $t-1$ and $t+1$, thus the temporal smoothing (TS) would significantly help in imputing the missing patches at

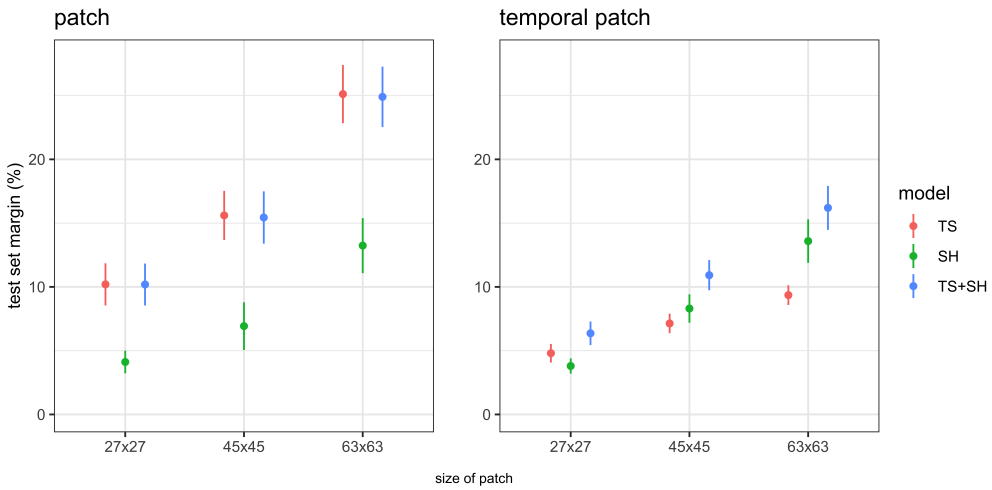


Figure 2.6: Numerical Analysis: random patch missing and temporal patch missing results. Three variants of our method are considered: TS, SH, and TS+SH. A detailed explanation is included in the main text. The scatter points show the average test set RSE margin over the baseline softImpute method, positive means performance better than softImpute. Error bar gives the 95% confidence interval.

t. However, when the location of the missing patch is highly auto-correlated, the missing patches in the matrices at $t-1, t$ and $t+1$ will have a large overlap[3], making temporal smoothing not as significant for improving the imputation as opposed to the previous scenario. Spherical harmonics, instead, can provide extra information over the whole missing patch, leading to more stable performance gains.

We now give a concrete example as an illustration of the numerical results shown in Figure 2.6. In Figure 2.7, we show the imputation made by the 4 models when we have 63×63 temporal patch missingness. It is easy to tell that softImpute barely imputes anything but a background value near zero in the patch. With temporal smoothing, the left and right borders of the patch are imputed. These are exactly the regions that are fully observed in their previous and next matrix. When using the spherical harmonics based auxiliary data, however, a more reasonable imputation is given in the patch.

With the numerical analyses, we have illustrated how different variations of our model can outperform softImpute, the baseline method. Since the missing pattern in the TEC map of the Madrigal database is mainly temporal patch missingness, it is expected that our extensions of the softImpute method can greatly help to impute this type of scientific image.

[3]In our design, each patch moves 6 columns/rows anti-clockwise per frame. So for a patch of any size, only 6 columns or rows of the missing patch can be observed in temporally adjacent matrices.

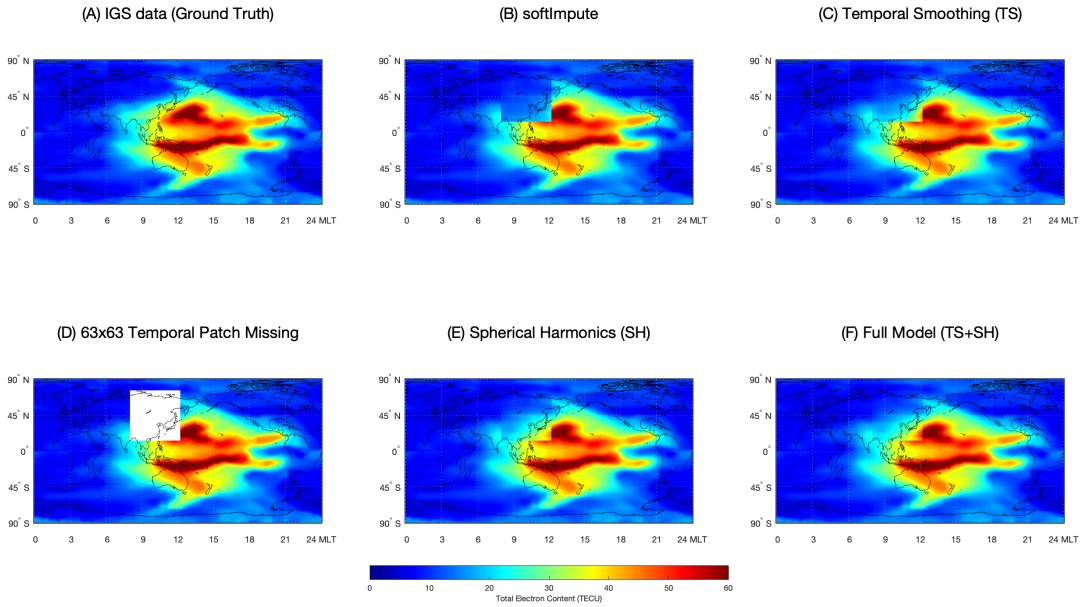


Figure 2.7: Example of imputing the IGS data with temporal patch missingness. (A) IGS data at 2017-09-08 02:15:00 UT. (B) Imputed with softImpute ($\lambda_1 = 0.9$). (C) Imputed with temporal smoothing ($\lambda_1 = 0.9, \lambda_2 = 0.05$). (D) 63×63 patch missingness. (E) Imputed with spherical harmonics auxiliary data ($\lambda_1 = 0.9, \lambda_3 = 0.01$). (F) Imputed with the full VISTA model ($\lambda_1 = 0.9, \lambda_2 = 0.05, \lambda_3 = 0.01$).

2.3.4 Methodology Comparison

Before concluding the numerical analysis section, we want to further compare our VISTA model against other competitive tensor completion methods beyond the SVD-based softImpute. We choose three methods: CP-WOPT (Acar et al., 2011), HaLRTC (Liu et al., 2012) and TMac (Xu et al., 2015). The CP-WOPT is trying to find the best low CP-rank decomposition of the imputation tensor to minimize the reconstruction error at the observed pixels. HaLRTC uses the Alternating Direction Method of Multipliers (ADMM) to estimate the imputation tensor to minimize the weighted nuclear norm of the matricized data tensor. TMac is trying to find the best low-rank factorization of the matricized data tensor and use the estimated factors to reconstruct the imputation.

For CP-WOPT, we set the $r = 10$ (rank of the imputation tensor based on CP decomposition). For HaLRTC, we set the $\alpha_n = \frac{1}{3}, n = 1, 2, 3$, so that each mode of the tensor gets equal weight in the objective function. For TMac, we set the targeting rank of each mode as 61 (latitude), 64 (magnetic local time), and 96 (temporal), respectively, based on the rank of the ground truth tensor after doing matricization along certain mode, and fit using a

rank-increasing strategy. All hyper-parameter choices can be considered fair because they are chosen based on either the default or the recommended approach. Alongside these models, we also fit VISTA with $\lambda_1 = 0.5, \lambda_2 = 0.2, \lambda_3 = 0.04$ and softImpute with $\lambda = 0.5$ based on the sequential tuning procedure described in Section 2.2.2. All simulation scenarios follow the previous setting and the test set RSE and error bar are shown in Figure 2.8.

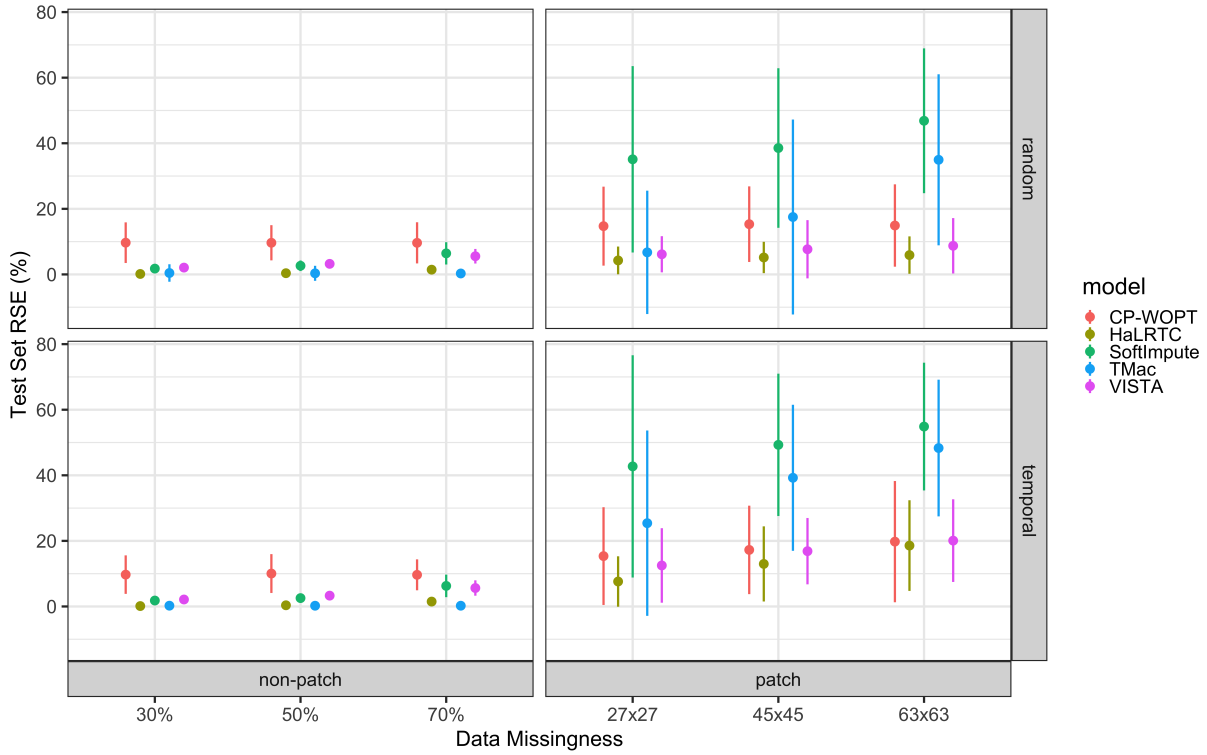


Figure 2.8: Method comparisons with CP-WOPT (Acar et al., 2011), HaLRTC (Liu et al., 2012), TMac (Xu et al., 2015), softImpute (Hastie et al., 2015) and our VISTA. All four simulation scenarios are tested: random, temporal, patch, and temporal patch. Three levels of data missingness are tested for each scenario. Choices of hyper-parameters are explained in texts. Error bar gives the 95% confidence interval.

Figure 2.8 shows that our method is competitive against other selected methods in all four simulation scenarios and across all three levels of data missingness. The method that comes close to our VISTA model is the HaLRTC, and both have short error bars, indicating their consistency across different frames. In Figure 2.9, we showed the test set RSE, by frame, for the simulation scenario of the temporal patch missingness.

We conclude from Figures 2.8 and 2.9 that our VISTA model is comparable to the HaLRTC method and better than other methods considered when there is temporal patch missingness, which is the dominant type of missingness in the real TEC data. Also, when com-

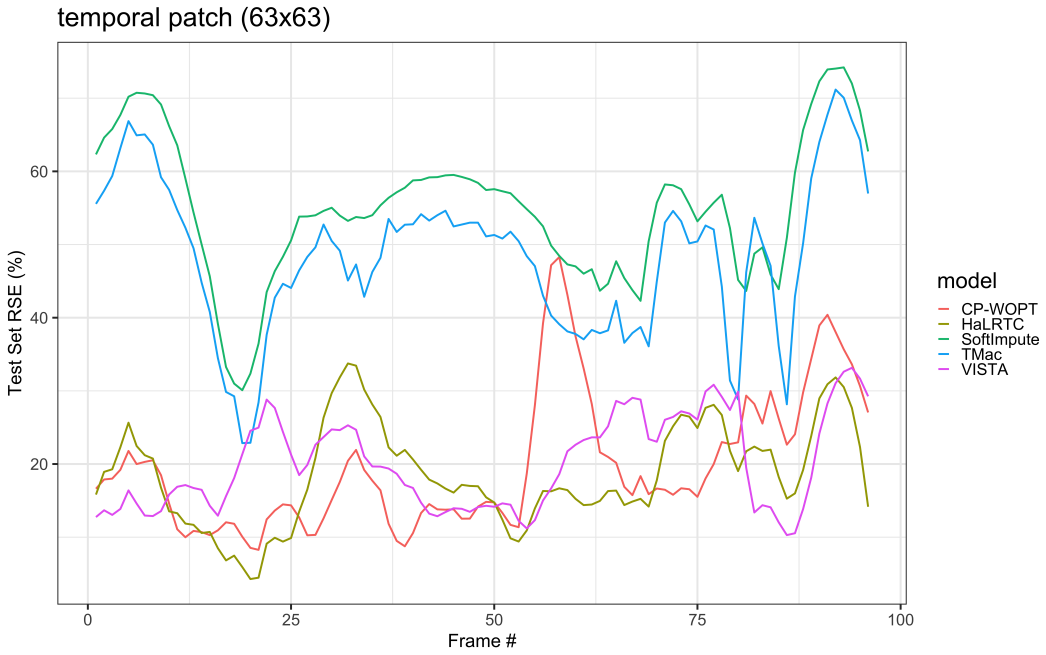


Figure 2.9: Method comparisons with CP-WOPT (Acar et al., 2011), HaLRTC (Liu et al., 2012), TMac (Xu et al., 2015), softImpute (Hastie et al., 2015) and our VISTA for the temporal patch scenario with a 63×63 box missing. The test set RSE is plotted against the frame number of the simulation data (96 frames in total).

pared to HaLRTC, each method has a few frames that do a better imputation. We recommend our approach for the ease of hyper-parameter tuning since based on our experience, VISTA is capable of providing reasonable imputations for a wide range of $\lambda_1, \lambda_2, \lambda_3$, and does not require information regarding the rank of the tensor at the first place. VISTA also allows for other auxiliary data to play a role, which has extra possibilities of improving the imputation. It is an encouraging result to see that by combining spherical harmonics with softImpute, one can bring the SVD-based method closer to or better than many competitive tensor completion methods, even given the bad performance of the baseline softImpute in the patch-missing scenarios.

There are limitations to our method too, especially in terms of computational efficiency. Take the temporal patch missingness (63×63) as an example, the HaLRTC takes 46.1 seconds to terminate, the CP-WOPT takes 144.2 seconds, the TMac takes 149.1 seconds, while our VISTA takes 220 seconds and there will be additional time consumed to generate the auxiliary data. All running times are recorded on a single-core (i9, 2.3 GHz), 16-GB memory (2400MHz, DDR4) CPU. Other simulation scenarios take a similar time to finish for each of the algorithms. We admit that our model is not as efficient and scalable as the others, but we want to highlight that our VISTA model can generate similar imputation

results for a wide range of tuning parameters, leading to potentially shorter running time if one counts the cross-validation step too.

In the next section, we present our empirical results on imputing the TEC map of the Madrigal database and address how our extensions can solve the problems of using softImpute for TEC map imputation, as discussed in Section 2.1.1.

2.4 TEC Map Reconstruction Results

We apply our model to reconstruct the TEC tensor in this section. For each day of the TEC data, our data has the form $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{m \times n \times T}$, where $m = 181$ corresponds to the latitude, $n = 361$ corresponds to the longitude and $T = 288$ corresponds to the number of time points of the day. We use TEC data after applying a 3×3 median filter in this section. Specifically, we chose two days of data for reconstruction demonstration: 2017-09-08 (a day with a geomagnetic storm) and 2017-09-03 (a non-storm day). During a storm day, the TEC map has more spatial structures and is more variable than that during a non-storm day.

Each whole day of the data contains 288 matrices, and in each matrix, we randomly drop 20% of the observed pixels and use them as the testing set. The remaining 80% of the data is used as the training set. We fit all 288 matrices together in one algorithm run and validate model performances on the testing set. In this section, the data follows the same pipeline as shown in Figure 2.2.

Hyper-parameters are selected based on the relative square error (RSE) on the randomly chosen validation set from the 80% of the training data. First we set $\lambda_2 = 0, \lambda_3 = 0$ and perform grid search on λ_1 to get the best λ_1 value. Again, this choice of λ_1 optimizes the performance of the original softImpute algorithm. Then, with our best λ_1 value, we perform a grid search on the temporal smoothing term and SH term separately for the best λ_2 and λ_3 values. Although this choice of the set of tuning parameters could be suboptimal, it is computationally much more efficient and serves the purpose of comparisons of various algorithms well.

Table 2.1 shows the results of our model on a storm day and a non-storm day. For the storm day, the best temporal smoothing hyper-parameter is $\lambda_2 = 0.2$ and the best spherical harmonic hyper-parameter is $\lambda_3 = 0.021$. The lowest RSE is achieved when we use the full model with $(\lambda_1, \lambda_2, \lambda_3) = (0.9, 0.2, 0.021)$. RSE reduces from 10.895% achieved by softImpute to 9.357% achieved by the full model. For the non-storm day, the best temporal smoothing hyper-parameter is $\lambda_2 = 0.31$ and the best spherical harmonic hyper-parameter is $\lambda_3 = 0.03$. When we use the full model with $(\lambda_1, \lambda_2, \lambda_3) = (0.9, 0.31, 0.03)$, we achieve

lowest RSE. RSE reduces from 10.424% achieved by softImpute to 8.592% achieved by the full model. Moreover, we also examined the mean-squared error (MSE) on the testing set. It turns out that the full model MSE also tends to be lower than the other models, and the softImpute model gives the highest MSE value among all the other imputation models we compare. The MSE of the non-storm day is overall lower than that of the storm day. This might be attributed to the relatively lower TEC values on a non-storm day. This also indicates that it is easier to impute non-storm day data, which varies less and has lower magnitudes. Based on the last two columns of Table 2.1, in over 97.5% of the time points, the TS, SH, and full model perform better than the softImpute result. In over 81% of the time points, the full model outperforms the other models, indicating that both temporal smoothing and auxiliary data from spherical harmonics can help improve the imputation results of TEC maps.

Storm Day (Sept 8, 2017)				
Model	test RSE	test MSE	# matrices better than softImpute	# matrices worse than Full model
softImpute	10.895%	2.675	/	285 (98.96%)
TS	9.643%	2.106	284 (98.62%)	267 (92.71%)
SH	9.936%	2.227	287 (99.65%)	274 (95.14%)
Full	9.357%	1.983	285 (98.96%)	/
Spherical Harmonics	17.354%	6.720	0 (0%)	288 (100%)
Non-Storm Day (Sept 3, 2017)				
Model	test RSE	test MSE	# matrices better than softImpute	# matrices worse than Full model
softImpute	10.424%	1.324	/	283 (98.26%)
TS	8.880%	0.958	281 (97.57%)	235 (81.60%)
SH	9.231%	1.032	287 (99.65%)	278 (96.53%)
Full	8.592%	0.895	283 (98.26%)	/
Spherical Harmonics	15.732%	2.893	0 (0%)	288 (100%)

Table 2.1: Empirical study results from the Madrigal database. The softImpute method has $\lambda_2, \lambda_3 = 0$. The TS method has $\lambda_3 = 0$ and has λ_2 the same as the full model. The SH method has $\lambda_2 = 0$ and has λ_3 the same as the full model.

Figures 2.10 and 2.11 show the original TEC median-filtered map, the map fitted with spherical harmonics (auxiliary data), and four imputed maps of a single time point of the storm day and the non-storm day data, respectively. The original softImpute method imposes a low-rank structure on the imputed matrix, leading to unreasonable gaps when being applied to scientific images. Our method with all three hyper-parameters reserves the features of the original plot, mitigates such gaps to allow for more temporal consis-

tency, and gives better spatial smoothness. For a non-storm day, the average rank of the imputed map with softImpute is 72, with temporal-smoothing is 98.5, and with the full model is 99.8. For a storm day, the average rank of the imputed map with softImpute is 85, with temporal smoothing is 104, and with the full model is 105.4. Note that Figures 2.10 and 2.11 have different scales due to the relatively lower TEC value on a non-storm day.

Comparing Figures 2.10 (D)(E) or (F)(C), we can see that by adding a temporal smoothing penalty, the undesirable low-rank structure is smoothed out. Comparing Figures 2.10 (D)(F) or (E)(C), the spherical harmonics help fill in missing patches caused by a lack of observations which appear in blue patches in (D) and (E) near the equator, i.e. 18 magnetic local time (MLT). The same patterns can be found in Figure 2.11 as well.

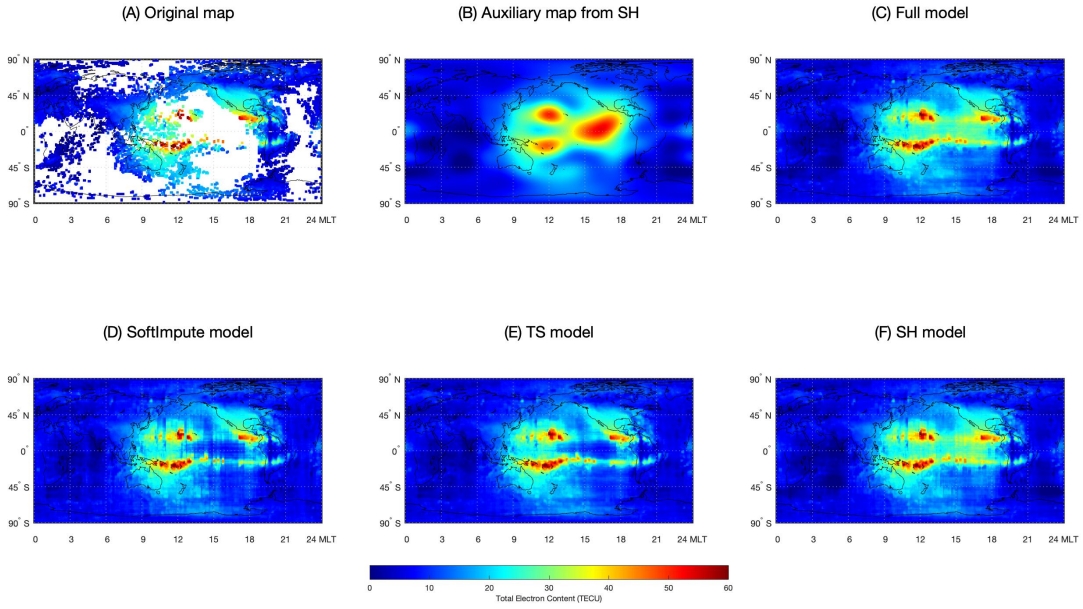


Figure 2.10: 2017-09-08/00:02:30 UT TEC maps. (A) Original median-filtered map. (B) Fitted TEC map by spherical harmonics. (C) Full model imputed map with $(\lambda_1, \lambda_2, \lambda_3) = (0.9, 0.2, 0.021)$. (D) SoftImpute fitted map with $\lambda_1 = 0.9$. (E) Temporal smoothing imputed map with $(\lambda_1, \lambda_2) = (0.9, 0.2)$. (F) SH imputed map with $(\lambda_1, \lambda_3) = (0.9, 0.021)$.

Overall, one can see that the temporal smoothing helps eliminate the low-rank structure; and the auxiliary data based on spherical harmonics, helps fill in large missing patches. These are exactly the two drawbacks of using the original trace-norm-based matrix completion method for imputing TEC maps that we mentioned in Section 2.1.

Just like the numerical analysis section, we also report the computational time for the real TEC map reconstruction here. For a single-core (i9, 2.3 GHz), 16-GB memory

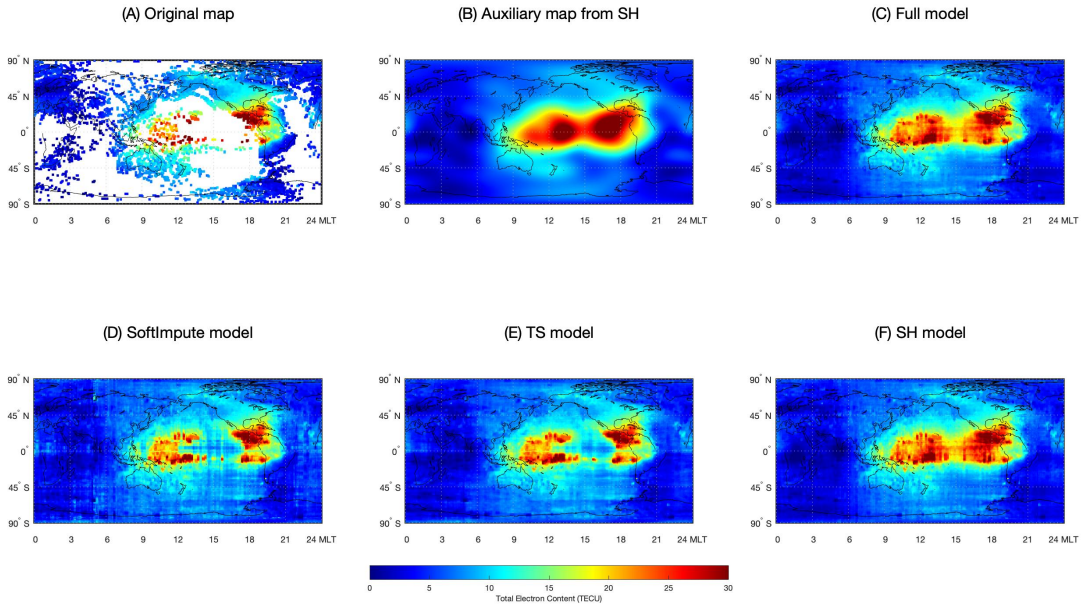


Figure 2.11: 2017-09-03/00:02:30 UT TEC maps. (A) Original median-filtered map. (B) Fitted TEC map by spherical harmonics. (C) Full model imputed map with $(\lambda_1, \lambda_2, \lambda_3) = (0.9, 0.31, 0.03)$. (D) SoftImpute fitted map with $\lambda_1 = 0.9$. (E) Temporal smoothing imputed map with $(\lambda_1, \lambda_2) = (0.9, 0.31)$. (F) SH imputed map with $(\lambda_1, \lambda_3) = (0.9, 0.03)$.

(2400MHz, DDR4) CPU, and for the storm day data$(181 \times 361 \times 288)$, the HaLRTC takes 112.3s, the CP-WOPT takes 1522.5s, the TMac takes 409.8s, while our VISTA model takes 1134.5s to terminate. The computational efficiency and scalability of our method are not ideal, but our current version is still a workable solution to the TEC map and similar video imputation problems.

Our work goes beyond methodology research and can contribute to generating high-quality TEC maps for the scientific field too. Meso-scale ionospheric structures, such as the channel-like TEC depletion in the north-south direction at 20 LT in Figure 1.2 and Figure 2.10 (i.e., equatorial plasma bubble [e.g., (Aa et al., 2018, 2019; Abdu, 2019) and references therein]) are associated with the largest navigation and communication satellite signal scintillation and thus of essential practical significance. However, this type of meso-scale structure can't be captured in the standard SH harmonics fitted maps, such as the IGS TEC maps. Therefore, the capability of preserving these meso-scale structures in the completed TEC map is a must for us to improve our specification and forecast of ionospheric space weather impact and will enable numerous studies in the domain field, such as the development and evolution of these meso-scale structures, their interaction

with large-scale TEC structures (i.e., the storm-enhanced density (Foster et al., 2005; Zou et al., 2013a, 2014; Zou and Ridley, 2016)), as well as the forecast of TEC using machine learning techniques based on these new maps. Our VISTA model gives an option to take a solid first step on these research tracks, as we briefly detail in the next section.

2.5 High-Resolution TEC Database with VISTA

As a product of this methodology research, we apply our VISTA algorithm extensively to the reconstruction of global TEC maps over a wide range of time. Our goal is to provide the scientific community with a high-resolution, high-cadence database that covers the past solar cycle and can preserve the meso-scale and large-scale structure of the Earth's ionosphere.

We omit most of the details for the construction of the database and refer our readers to our published paper (Sun et al., 2023a) and in this section, we summarize the database at a high level. The database, as compared to the existing IGS TEC database, features a higher spatial resolution at 1°-latitude by 1°-longitude, and a higher temporal resolution at 5-min. The database spans 16 years from 2005 to 2020. We summarize the entire workflow for constructing the database in Figure 2.12.

To tune the parameter of the VISTA algorithm, including $\lambda_1, \lambda_2, \lambda_3$ in the optimization problem (2.3) and $l_{\max}, \nu$ for the spherical harmonics, we divide the 16-year period into four sub-intervals: $2005 \sim 2011$, $2012 \sim 2014$, $2015 \sim 2018$, $2019 \sim 2020$. Within each interval, we pick one month of data to tune the parameters and consequently, all days within each interval share the same set of tuning parameters. The partition is chosen as such since these four intervals have relatively different data missing percentages in the raw Madrigal TEC data, where the missing percentages are $> 90\%$ for $2005 \sim 2011$, $\sim 85\%$ for $2012 \sim 2014$, $\sim 82\%$ for $2015 \sim 2018$ and $< 80\%$ for $2019 \sim 2020$. The four months are 2009-Apr, 2014-Jan, 2015-Sep, and 2019-May, which are chosen to cover different geomagnetic activity levels and different periods in the previous solar cycle. We list the final choice of tuning parameters in Table 2.2. We visualize all critical maps in the pipeline in Figure 2.13.

To validate the database using an independent source of TEC measurements, we follow the validation approach used by the IGS database (Hernández-Pajares et al., 2009, 2020) and use the TEC measurements from the dual frequency altimeters on board the JASON satellite series as the reference TEC level. We use the JASON-1, JASON-2, and JASON-3 TEC data from the Madrigal database (MIT Haystack Observatory, 2012) as the source of reference TEC data. These TEC measurements are based on satellites that are not mea-

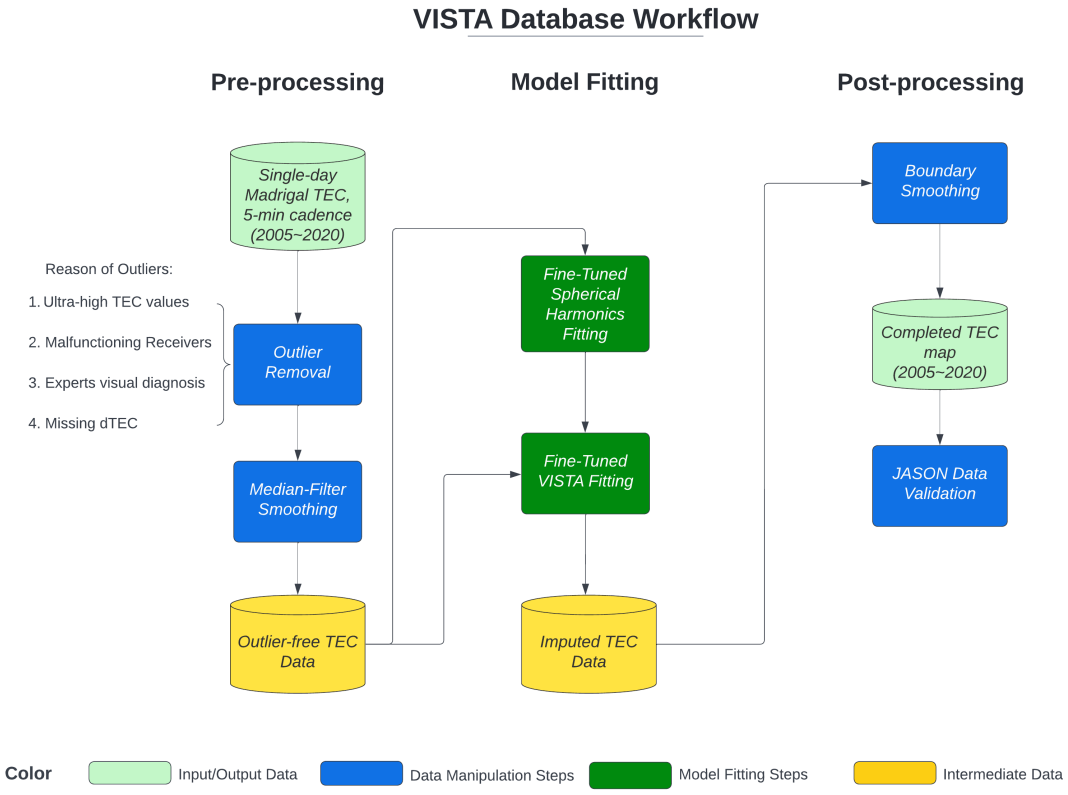


Figure 2.12: Complete data generating workflow. The source data is the Madrigal TEC data containing missing values. We fit the spherical harmonics smoothing algorithm with ℓ_2 regularization to the source data, after removing outliers, to generate the auxiliary data. Combining both the source and the auxiliary data, we run the VISTA algorithm to generate the complete, low-rank, and locally smoothed TEC map (the imputed TEC data). Finally, we run a moving average smoother to smooth the completed TEC maps near the day-to-day boundary to remove the impact introduced by daily fluctuations. More details on the VISTA fitting are included in Figure 2.2.

Category	Notation	$2005 \sim 2011$	$2012 \sim 2014$	$2015 \sim 2018$	$2019 \sim 2020$
VISTA	λ_1	0.3	0.2	0.2	0.3
	λ_2	0.25	0.40	0.40	0.25
	λ_3	0.12	0.15	0.12	0.12
SH	l_{max}	6	9	7	7
	v	0.1	0.1	0.1	0.1

Table 2.2: Final tuning parameter choices for constructing the VISTA database for years in the four intervals: $2005 \sim 2011$, $2012 \sim 2014$, $2015 \sim 2018$, $2019 \sim 2020$.

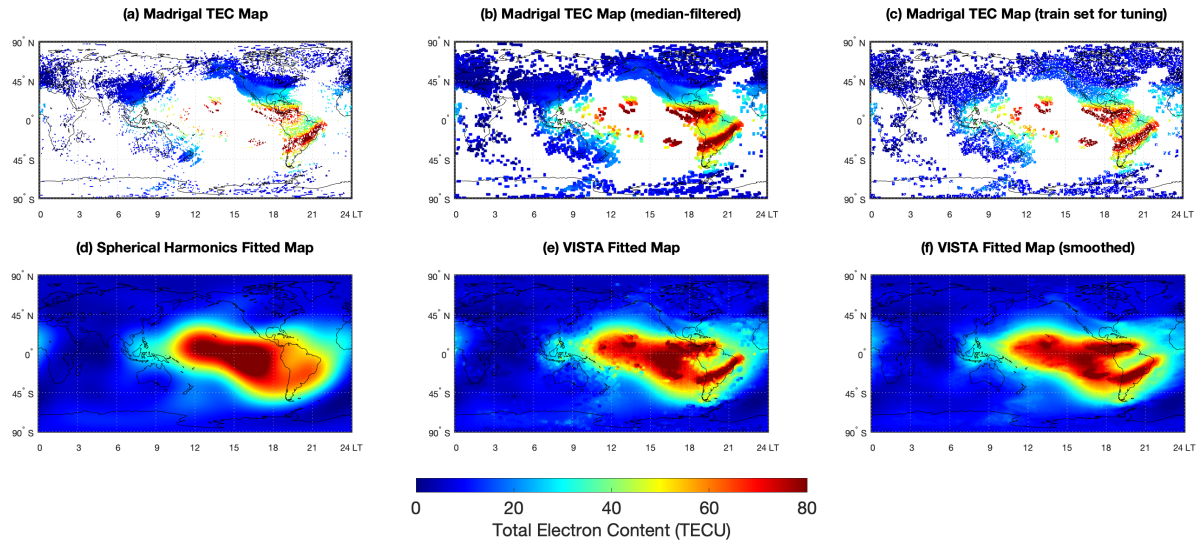


Figure 2.13: All critical TEC-related maps in our data pipeline, with the sample being the last frame (23:57:30 UT) of March 17, 2015. (A) shows the raw Madrigal TEC map after outlier removal. (B) shows the raw Madrigal TEC map processed by the median filter, which is the input data of our SH and VISTA algorithms. (C) shows the training set ($\sim 80\%$ of the observed pixels in (B)) when we do parameter tuning. (D) is the spherical harmonics (SH) map, fitted with $l_{max} = 7, v = 0.1$ using (B). (E) shows the VISTA map using (B) and (D), with $\lambda_1 = 0.2, \lambda_2 = 0.40, \lambda_3 = 0.12$. (F) shows the smoothed version of (E) when we apply day-to-day boundary smoothing.

sured in the same way as the Madrigal TEC and can thus serve as an external validation benchmark.

We compare the data collected by the JASON satellites with the corresponding data in our VISTA output during 2005-2020 by converting the JASON TEC measurements into 1°latitude $\times$ 1°longitude spatial resolution with 5-minute cadence, which has the same spatio-temporal resolution as our VISTA TEC. Each JASON TEC record is assigned to its nearest neighbor in the spatio-temporal grid based on the resolution specified above. Then we calculate the difference of the TEC value measured by JASON and VISTA to get the residuals and group the residuals by year. We apply the same procedure to the median-filtered Madrigal TEC as well since it is the source data of VISTA. Additionally, we adjust the inter-satellite bias among the three JASON satellites based on the bias estimation in Table 5 of Azpilicueta and Nava (2021). Specifically, we subtract a constant of 3.5 TECu from all JASON-2 TEC measurements and 1.0 TECu from all JASON-3 measurements to make their TEC scale on par with that of JASON-1.

Figure 2.14 shows the mean and standard deviation of the yearly residual for both the Madrigal TEC and our VISTA database. We differentiate the pixels in the VISTA TEC

based on whether the pixel has the original Madrigal TEC observation. All pixels of VISTA TEC with Madrigal TEC observations are labeled as "(Madrigal = A)", and "(Madrigal = NA)" is used to denote the remaining pixels of VISTA TEC without the corresponding Madrigal TEC observations. One can see that the bias of the database, compared to the JASON satellite TEC measurements, shows a very similar trend for both the Madrigal TEC and VISTA TEC when the Madrigal data is available (i.e. Madrigal = A), and the bias gets slightly larger by $0.5 \sim 1$ TECu when no Madrigal TEC is available during the fitting. The trend is similar in the standard deviation of the residual. The yearly coverage of different JASON satellites is shown on top of each panel. During years $2010 \sim 2016$, we see relatively higher bias and standard deviation of the residual.

Compared to the JASON validation results of the IGS TEC maps (Hernández-Pajares et al., 2009), the VISTA database shows a lower mean and standard deviation of the bias, though the validation period does not coincide exactly. The IGS TEC map has an average bias of 1.00 TECu and the standard deviation of the bias is around 4.42 TECu, during the validation period of $2002 \sim 2007$. The VISTA database, on the other hand, shows an average bias around $-0.3 \sim 0.5$ TECu and a standard deviation around or below 4 TECu for the period $2005 \sim 2007$. We conclude that our new database has higher imputation accuracy as compared to the existing database and also features high spatio-temporal resolutions. The complete TEC map database can be used for various ionospheric physics and space weather applications. See our published papers (Zou et al., 2021; Wang et al., 2023) as examples of using the VISTA TEC database.

2.6 Conclusion

In this paper, we proposed the Video Imputation with SoftImpute, Temporal smoothing and Auxiliary data (VISTA) method, which gives two extensions of the softImpute algorithm for matrix completion in reconstructing time series of matrices with spatial dimensions. The incorporation of auxiliary data and temporal smoothness via penalty terms in the loss function enables us to combine external information and achieve temporal information sharing. The proposed algorithm is implemented in R and made available on GitHub. We prove theoretical properties of the algorithm such as the convergence rate as in the original softImpute paper. In our numerical simulations that mimic the real-world scenario, the proposed method works out very well and vastly improves the performance of the SVD-based method. The real data analysis further demonstrates satisfactory performance.

There are a few extensions of our method that may trigger further research interests.

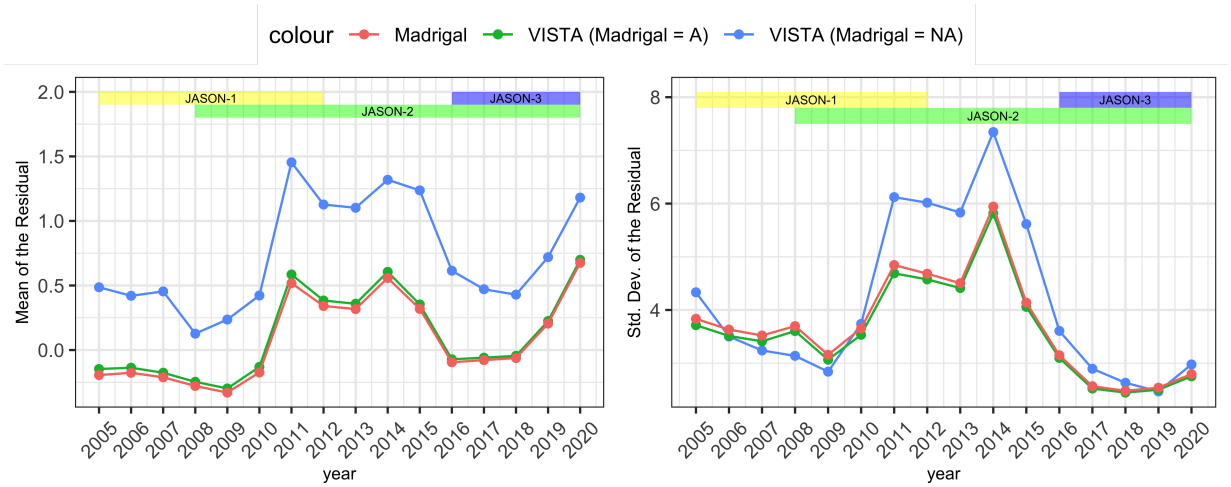


Figure 2.14: Residual mean and standard deviation, data grouped by year. Three types of data points are considered: the Madrigal TEC, the VISTA TEC with Madrigal observation (Madrigal = A), and the VISTA TEC without Madrigal observation (Madrigal = NA). The time spans that each JASON satellite provides the validation data are shown as colored bars on top. Inter-satellite biases are corrected based on Azpilicueta and Nava (2021) to make the TEC measurements from JASON-2 and JASON-3 on par with those from JASON-1. On average, each year has $10^{5.8}$ validation pixels.

For instance, when one creates the majorization bound, more generally, one can:

$$\|\mathrm{P}_{\boldsymbol{\Omega}_t}(\mathbf{X}_t - \mathbf{A}_t(\mathbf{B}_t^{(k)})^\top)\|_{\mathrm{F}}^2 \leq \|\mathrm{P}_{\boldsymbol{\Omega}_t}(\mathbf{X}_t) + a \cdot \mathrm{P}_{\boldsymbol{\Omega}_t^\perp}(\mathbf{A}_t^{(k)}(\mathbf{B}_t^{(k)})^\top) - \mathbf{A}_t(\mathbf{B}_t^{(k)})^\top\|_{\mathrm{F}}^2. \tag{2.20}$$

We found that by letting $a < 1$, the algorithm still converges and the smaller a is, the quicker the algorithm converges. However, given the same tuning parameter, $a = 1$ still has better empirical performance on real TEC data. But this still suggests more room for new research on the majorization bound to further improve these types of methods.

The output of the imputed TEC data suggests that the rank of the imputed map, for a single frame, is around $80 \sim 120$, while the maximum possible rank should be 181. This suggests that the true underlying map can have a low-rank nature, which can be modeled more directly via a non-convex matrix completion method. We adopt the softImpute-ALS for its bi-convexity and its flexibility to add temporal smoothness and auxiliary data penalty. More research on how to combine these penalties with a non-convex matrix or tensor completion framework can be beneficial, especially when one has prior knowledge about the low-rank nature of the data.

Another extension of our framework is to localize the temporal smoothness and auxiliary data penalty to a specific spatial-temporal patch. Our current setup has penalties

for all frames and all pixels, which can be sub-optimal. We found that the optimal tuning parameters for storm and non-storm days differ, which suggests that different times can have different temporal/spatial smoothness, indicating the significance of a more localized spatial-temporal smoothness penalty. More research on this thread to further extend our framework and similar tensor completion methods can be highly beneficial.

The last extension is about the selection and generation of auxiliary data. In this chapter, we adopt the spherical harmonics as the auxiliary data and fit it separately from our main model. One can also select the outputs of other imputation methods (e.g., HaLRTC, CP-WOPT, etc.) as the auxiliary data alongside the spherical harmonics. The selection should be based on the real data. Spherical harmonics is a good choice for the TEC data but is not necessarily good for other types of data. Also, one can fit the auxiliary data jointly with the imputation model. We do not do it here since more tuning parameters will get involved if one fits the imputation model jointly with the auxiliary data generation, but it is possible to generate the auxiliary data together with the imputation map.

The newly proposed algorithms are targeted but not restricted to the temporal TEC map reconstruction. We hope that the proposed algorithm can serve a larger scientific community and stimulate more interest from statisticians for more thorough theoretical investigations.

Codes for this chapter can be found at: https://github.com/husun0822/TEC_impute. An interactive website for the VISTA algorithm and VISTA TEC database is at: https://vista-tec.shinyapps.io/VISTA-Dashboard/. The VISTA TEC database is at: https://deepblue.lib.umich.edu/data/concern/data_sets/nc580n00z?locale=en.

CHAPTER 3

Conformalized Tensor Completion with Riemannian Optimization

3.1 Introduction

Tensor, or multi-dimensional array, has become a popular data format in several applications such as collaborative filtering (Bi et al., 2018), financial time series modeling (Li and Xiao, 2021), hypergraph networks analysis (Ke et al., 2019), neuroimaging study (Li et al., 2018) and astrophysics imaging analysis (Sun et al., 2023b). Tensor gains this popularity due to its efficient representation of structural high-dimensional data. For example, in collaborative filtering (Bi et al., 2018), the rating data is naturally embedded in an order-3 tensor with user$\times$item$\times$context with each entry being the rating by a user on a certain item under a specific context. In neuroimaging analysis (Wei et al., 2023), as another example, each brain voxel in the order-3 tensor is identified by its coordinate in the 3-D Euclidean space.

Tensor completion (Yuan and Zhang, 2016; Xia et al., 2021; Cai et al., 2022a) is a technique that provides an estimator of the tensor when missing values are present. Typically, given only one tensor sample with missingness, tensor completion aims at finding a low-rank tensor that best imputes the missing entries. Various optimization techniques (Kressner et al., 2014; Yuan and Zhang, 2016; Lee and Wang, 2020; Cai et al., 2022a) have been proposed for computationally efficient tensor completion and the statistical error of tensor completion has also been carefully investigated (Xia et al., 2021).

However, given the progress above, very little work has been done on the uncertainty quantification of tensor completion. Existing work on the uncertainty quantification of matrix completion (Carpentier et al., 2018, 2019; Chen et al., 2019c; Xia and Yuan, 2021; Farias et al., 2022) and tensor completion (Cai et al., 2022b) typically relies on asymptotic analysis of the estimator by a specific completion algorithm and assumes that data is missing uniformly at random. In this chapter, we aim to devise a data-driven approach

that does not rely on a specific choice of the completion algorithm nor assume the data is missing uniformly at random, which is more adaptive to real application scenarios.

Conformal prediction (Vovk et al., 2005) is a model-agnostic approach for uncertainty quantification. Recently, Gui et al. (2023) applies the idea of conformal prediction to matrix completion under the assumption that data is missing independently. The method requires one to estimate the missing propensity of each matrix entry and weigh them accordingly to construct well-calibrated confidence regions. In this chapter, we generalize this idea to tensor completion. The generalization is non-trivial as one cannot simply reshape the tensor back to a matrix for the conformal prediction without significantly increasing the dimensionality of the nuisance parameter. We keep the tensor structure and leverage low-rank tensor representations for dimension reduction. Furthermore, we do not assume data is missing independently but allow for locally-dependent missingness. We capture such correlatedness of missingness by a novel low-rank tensor Ising model, which could be of independent interest. Finally, we propose a Riemannian gradient descent algorithm (Kressner et al., 2014) for scalable computation, which is necessary since tensor data is typically high-dimensional.

The key insight of the method is that one puts higher weight on the tensor entries with a higher probability of missing, which can be considered as "nearest neighbors" of the missing entries. Such a weighted conformal prediction approach (Tibshirani et al., 2019) is also seen in spatial conformal prediction (Mao et al., 2022) and localized conformal prediction (Guan, 2023) where higher weights are put on neighbors in the Euclidean or feature space. However, our method is significantly different in that we estimate the weights by using the entire tensor and determine the weights of all tensor entries altogether while other methods determine the weight of each data locally and thus can be slow under the tensor setting.

There are plenty of other literature that are relevant to the uncertainty quantification of matrix and tensor completion, beyond the ones mentioned above. Alquier (2015) and Yuchi et al. (2023) utilize a Bayesian framework for matrix completion that comes with automatic uncertainty quantification. Adopting a Bayesian framework, however, still requires distribution assumptions over the data-generating process and can be computationally inefficient when scaled to tensors. Matrix completion by deep learning (Zeldes et al., 2017; Kasalicky et al., 2023) can output uncertainty quantification in a model-agnostic and distribution-free fashion but lacks theoretical coverage guarantee. One recent work on matrix prediction with conformal prediction (Shao and Zhang, 2023) considers the uncertainty quantification for new rows and columns of the data matrix when the matrix (e.g., adjacency matrix for a network) is expanding. This setup is different from ours in

that our tensor data is static and the data missingness is dependent among neighboring tensor entries, which mimics the missing pattern for spatio-temporal data. In general, to the best of our knowledge, our work is the first to consider uncertainty quantification for tensor completion with locally dependent data missingness via the conformal prediction approach that has a coverage guarantee.

The remainder of the chapter is organized as follows. Section 3.2 describes the conformalized tensor completion (CTC) method and the probabilistic model for the data missingness. Section 3.3 is dedicated to the computational algorithm of the CTC. We conduct theoretical analyses of our model in Section 3.4. We validate the performance of our proposed CTC using extensive simulations in Section 3.5 and a real data application to the TEC reconstruction problem in Section 3.6. Section 3.7 concludes. The Appendix B contains technical proofs and additional details and results of the simulation and data application.

3.2 Method

Most of the notations used in this chapter follow the definitions in Section 1.3 and here we make a few additional remarks on the additional notations used. For a K-mode tensor with size $d_1 \times \cdots \times d_K$, we use $\mathbb{S}$ to denote $[d_1] \times \cdots \times [d_K]$, namely the indices of all tensor entries, and we often use a single index such as i, j, s instead of a K-tuple to denote elements from $\mathbb{S}$ for notational brevity.

In this chapter, when referring to a tensor that is a random variable, we add a tilde over the top of the tensor such as $\widetilde{\boldsymbol{\mathcal{W}}}, \widetilde{\boldsymbol{\mathcal{X}}}$ and use the raw version $\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{X}}$ to denote concrete samples. We add an asterisk to the superscript such as $\boldsymbol{\mathcal{X}}^*, \boldsymbol{\mathcal{B}}^*$ to denote the non-random, ground truth parameters.

Suppose we have a K-mode random tensor $\widetilde{\boldsymbol{\mathcal{X}}}$ of size $d_1 \times \cdots \times d_K$. Further, suppose that one obtains a sample $\boldsymbol{\mathcal{X}}$ for $\widetilde{\boldsymbol{\mathcal{X}}}$ with part of the entries in $\boldsymbol{\mathcal{X}}$ missing. To encode the missingness in $\boldsymbol{\mathcal{X}}$, we define the binary missingness tensor $\boldsymbol{\mathcal{W}} \in \{-1, 1\}^{d_1 \times \cdots \times d_K}$ and set $\boldsymbol{\mathcal{W}}_s = 1$ when $\boldsymbol{\mathcal{X}}_s$ is observed and $\boldsymbol{\mathcal{W}}_s = -1$ when $\boldsymbol{\mathcal{X}}_s$ is missing. We assume that the missingness $\boldsymbol{\mathcal{W}}$ is a sample of a random binary tensor $\widetilde{\boldsymbol{\mathcal{W}}}$ whose likelihood is $p(\cdot)$.

The tensor completion problem (Yuan and Zhang, 2016; Xia et al., 2021; Cai et al., 2022a) deals with estimating the values in $\boldsymbol{\mathcal{X}}$ where $\boldsymbol{\mathcal{W}}_s = -1$, i.e. where data is missing. Although the main framework of this chapter does not rely on a specific choice of the tensor completion algorithm, it is beneficial to provide one example here which is also the algorithm we will be using in our numerical experiments and data application.

Since one only has one sample $\boldsymbol{\mathcal{X}}$ of $\widetilde{\boldsymbol{\mathcal{X}}}$, estimating the missing values in $\boldsymbol{\mathcal{X}}$ is impossible

without imposing additional parsimony over the estimator. Following the literature on tensor completion (Kressner et al., 2014; Xia et al., 2021; Cai et al., 2022d), we assume that the estimator has a low tensor rank and solve for the estimator by the following constrained least-square problem:

$$\min_{\boldsymbol{\mathcal{A}}:\mathrm{rank}(\boldsymbol{\mathcal{A}})\leq \mathbf{r}} \quad \frac{1}{2}\sum_{s:\boldsymbol{\mathcal{W}}_s=1}(\boldsymbol{\mathcal{X}}_s-\boldsymbol{\mathcal{A}}_s)^2\,, \tag{3.1}$$

where the notion of tensor rank will be introduced later. We denote the minimizer of (3.1) as $\widehat{\boldsymbol{\mathcal{X}}}$. The goal of this chapter is to quantify the uncertainty for $\widehat{\boldsymbol{\mathcal{X}}}$ by constructing confidence interval $C(\widehat{\boldsymbol{\mathcal{X}}})$ around $\widehat{\boldsymbol{\mathcal{X}}}$ to cover $\boldsymbol{\mathcal{X}}$ with a pre-specified level of confidence. The framework, called conformalized tensor completion, will be introduced next.

3.2.1 Conformalized Tensor Completion (CTC)

Conformal prediction (Vovk et al., 2005) is a model-agnostic, distribution-free approach for predictive uncertainty quantification. To put in the context of the tensor completion problem, we utilize specifically the *split conformal prediction* (Papadopoulos et al., 2002) approach for its simplicity and scalability to complex data structures such as tensor data. We leave the discussion of *full conformal prediction* (Shafer and Vovk, 2008) to future work.

Split conformal prediction starts by partitioning all observed entries in $\boldsymbol{\mathcal{X}}$, whose indices are denoted as $\mathbb{S}_{obs}$, randomly into a training set $\mathbb{S}_{tr}$ and a calibration set $\mathbb{S}_{cal}$. One first provides a tensor completion estimator $\widehat{\boldsymbol{\mathcal{X}}}$ using the training set *only*, say by solving for (3.1) using entries in $\mathbb{S}_{tr}$. Then one calculates the *non-conformity score* over the calibration set by a score function $\mathcal{S}(\boldsymbol{\mathcal{X}}_s,\widehat{\boldsymbol{\mathcal{X}}}_s)$ such as $\mathcal{S}(\boldsymbol{\mathcal{X}}_s,\widehat{\boldsymbol{\mathcal{X}}}_s)=|\boldsymbol{\mathcal{X}}_s-\widehat{\boldsymbol{\mathcal{X}}}_s|$. To quantify the uncertainty of $\widehat{\boldsymbol{\mathcal{X}}}_{s^*}$ at any missing entry $s^*\in\mathbb{S}_{miss}$, where $\mathbb{S}_{miss}$ includes the indices of all missing entries, the canonical conformal interval at $(1-\alpha)$ confidence level is constructed as $C_{1-\alpha,s^*}(\widehat{\boldsymbol{\mathcal{X}}})=\{x\in\mathbb{R}|\mathcal{S}(x,\widehat{\boldsymbol{\mathcal{X}}}_{s^*})\leq\widehat{q}\}$, with $\widehat{q}$ defined as:

$$\widehat{q}=\mathcal{Q}_{1-\alpha}\left(\frac{1}{|\mathbb{S}_{cal}|+1}\cdot\sum_{s\in\mathbb{S}_{cal}}\delta_{\mathcal{S}(\boldsymbol{\mathcal{X}}_s,\widehat{\boldsymbol{\mathcal{X}}}_s)}+\frac{1}{|\mathbb{S}_{cal}|+1}\cdot\delta_{+\infty}\right), \tag{3.2}$$

where δ_a is a point mass at $x=a$ and $\mathcal{Q}_\tau(\cdot)$ extracts the $(100\tau)^{\text{th}}$ quantile of a CDF. The validity of such a conformal interval $C_{1-\alpha,s^*}(\widehat{\boldsymbol{\mathcal{X}}})$ relies on the assumption of *data exchangeability* (Lei et al., 2018). To put it in the context of tensor completion, we re-label $\mathbb{S}_{cal}\cup\{s^*\}$ as $\{s_1,\ldots,s_{n+1}\}$, with $n=|\mathbb{S}_{cal}|$ and $s_{n+1}=s^*$ and define event $\boldsymbol{\mathcal{E}}_0$ as:

$$\boldsymbol{\mathcal{E}}_0=\left\{\widetilde{\boldsymbol{\mathcal{W}}}_s=1 \text{ for } s\in\mathbb{S}_{tr}\cup\mathbb{S}_{cal},\mathbb{S}_{cal}\cup\{s^*\}=\{s_1,\ldots,s_{n+1}\} \text{ and } \widetilde{\boldsymbol{\mathcal{W}}}_s=-1 \text{ o.w.}\right\}. \tag{3.3}$$

Then data exchangeability is equivalent to saying that the probability:

$$\mathrm{P}\left[\widetilde{\boldsymbol{\mathcal{W}}}_{s_k} = -1 \text{ and } \widetilde{\boldsymbol{\mathcal{W}}}_s = 1 \text{ for } s \in \mathbb{S}_k \middle| \boldsymbol{\mathcal{E}}_0\right]$$

is equal for all $k = 1, \ldots, n+1$, where $\mathbb{S}_k = \{s_1, \ldots, s_{n+1}\} \backslash \{s_k\}$. Equivalently, this states that conditioning on observing data only from $\mathbb{S}_{tr}$ and n out of $n+1$ entries from $\{s_1, \ldots, s_{n+1}\}$, it is equally likely to observe any n entries from $\{s_1, \ldots, s_{n+1}\}$. This assumption will hold when data are missing independently with the same probability or uniformly at random in short, a common assumption made in the literature on matrix/tensor completion uncertainty quantification (Chen et al., 2019c; Cai et al., 2022b). However, this assumption might not hold when the data missingness is dependent and is surely violated when the missingness is independent but with heterogeneous probabilities. Therefore, it is necessary to account for more general data missing patterns when conducting the uncertainty quantification.

We modify the canonical conformal prediction to accommodate more general data missing patterns by re-weighting each calibration entry using the weighted exchangeability framework (Tibshirani et al., 2019). The result is summarized in Proposition 3.2.1.

Proposition 3.2.1. *For any testing entry $s^* \in \mathbb{S}_{miss}$, let $\mathbb{S}_{cal} \cup \{s^*\} = \{s_1, \ldots, s_{n+1}\}$ and $\mathbb{S}_k = \{s_1, \ldots, s_{n+1}\} \setminus \{s_k\}$, then define p_k as:*

$$p_k = \mathrm{P}\left(\widetilde{\boldsymbol{\mathcal{W}}}_s = 1 \text{ for } s \in \mathbb{S}_{tr} \cup \mathbb{S}_k, \widetilde{\boldsymbol{\mathcal{W}}}_s = -1 \text{ o.w.}\right), \tag{3.4}$$

for $k = 1, \ldots, n+1$. Let $\widehat{\boldsymbol{\mathcal{X}}}$ be the output of any tensor completion method using entries only from $\mathbb{S}_{tr}$ and define $\widehat{q}_{s^*}$ as:

$$\widehat{q}_{s^*} = \mathcal{Q}_{1-\alpha}\left(\sum_{i=1}^{n} \omega_{s_i} \cdot \delta_{\mathcal{S}(\boldsymbol{\mathcal{X}}_s, \widehat{\boldsymbol{\mathcal{X}}}_s)} + \omega_{s_{n+1}} \cdot \delta_{+\infty}\right), \quad \text{where } \omega_k = \frac{p_k}{\sum_{i=1}^{n+1} p_i}, \tag{3.5}$$

and construct the $(1-\alpha)$-level conformal interval as $C_{1-\alpha,s^*}(\widehat{\boldsymbol{\mathcal{X}}}) = \{x \in \mathbb{R} | \mathcal{S}(x, \widehat{\boldsymbol{\mathcal{X}}}_{s^*}) \leq \widehat{q}_{s^*}\}$, then given the definition of $\boldsymbol{\mathcal{E}}_0$ in (3.3), we have:

$$\mathrm{P}\left(\boldsymbol{\mathcal{X}}_{s^*} \in C_{1-\alpha,s^*}(\widehat{\boldsymbol{\mathcal{X}}}) \middle| \boldsymbol{\mathcal{E}}_0\right) \geq 1 - \alpha. \tag{3.6}$$

We provide the detailed proof in Appendix B.1.1. Proposition 3.2.1 indicates that as long as one can properly weight each calibration entry in proportion to p_k as defined in (3.4), one can obtain the conditional coverage guarantee in (3.6). A similar result to

Proposition 3.2.1 has been established for conformalized matrix completion (Gui et al., 2023), where the data is assumed to be missing independently. In this chapter, we do not assume independent missingness but provide a more general statement that requires one to weight each calibration and testing entry by directly evaluating the likelihood of $\widetilde{\boldsymbol{\mathcal{W}}}$ under $n+1$ different missingness, where each time we set 1 out of $n+1$ entries as missing. In Section 3.2.2, we will formally introduce the likelihood of the binary tensor $\widetilde{\boldsymbol{\mathcal{W}}}$ that nests the independent missingness as a special case.

3.2.2 Missing Propensity Model

The key to constructing the conformal interval with coverage guarantee is to properly weight each calibration sample by p_k in (3.4), which requires the knowledge of the likelihood of $\widetilde{\boldsymbol{\mathcal{W}}}$. In practice, one does not have access to such knowledge but needs to estimate the likelihood of $\widetilde{\boldsymbol{\mathcal{W}}}$, given a single sample $\boldsymbol{\mathcal{W}}$, and then plug in (3.4) to get an estimator $\widehat{p}_k$. Previous works (Chen et al., 2019c; Cai et al., 2022b; Gui et al., 2023) assume that all matrix/tensor entries are missing independently, potentially with heterogeneous probabilities. This assumption, however, is not general enough. For example, for spatio-temporal tensors, data might miss together if located close in space or time. As another example, hypergraph adjacency tensor (Ke et al., 2019) may have data missing together if two entries share a group of nodes in the network.

Accounting for the dependencies of binary random variables turns out to be even more challenging in our context because all the binary random variables in $\widetilde{\boldsymbol{\mathcal{W}}}$ are embedded in a tensor grid with ultra-high dimensionality. Fortunately, the Ising model (Cipra, 1987) provides one way of modeling dependent binary random variables on a lattice grid. The binary random variables here are the indicators of data missingness instead of atomic spins in ferromagnetism but a similar idea applies to our modeling context.

To start with, the Ising model prescribes a Boltzmann distribution for $\widetilde{\boldsymbol{\mathcal{W}}}$: $p(\widetilde{\boldsymbol{\mathcal{W}}}) \propto \exp[-\beta\mathcal{H}(\widetilde{\boldsymbol{\mathcal{W}}})]$, where $\beta > 0$ is the inverse temperature parameter and $\mathcal{H}(\widetilde{\boldsymbol{\mathcal{W}}})$ is the *Hamiltonian* of $\widetilde{\boldsymbol{\mathcal{W}}}$, describing the "energy" of $\widetilde{\boldsymbol{\mathcal{W}}}$. In this chapter, we extend the richness of this model by augmenting $p(\widetilde{\boldsymbol{\mathcal{W}}})$ with an unknown tensor parameter $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$ such that:

$$p(\widetilde{\boldsymbol{\mathcal{W}}}|\boldsymbol{\mathcal{B}}) \propto \exp\{-\mathcal{H}(\widetilde{\boldsymbol{\mathcal{W}}}|\boldsymbol{\mathcal{B}})\}, \tag{3.7}$$

$$\mathcal{H}(\widetilde{\boldsymbol{\mathcal{W}}}|\boldsymbol{\mathcal{B}}) = -\frac{1}{2}\sum_{i\sim j} g(\boldsymbol{\mathcal{B}}_i, \boldsymbol{\mathcal{B}}_j)\widetilde{\boldsymbol{\mathcal{W}}}_i\widetilde{\boldsymbol{\mathcal{W}}}_j - \sum_i h(\boldsymbol{\mathcal{B}}_i)\widetilde{\boldsymbol{\mathcal{W}}}_i, \tag{3.8}$$

where $i, j \in [d_1] \times \cdots \times [d_K]$, $g(\cdot,\cdot)$ is a symmetric bi-variate function and $h(\cdot)$ is a univariate function with the inverse temperature parameter β being incorporated into $g(\cdot,\cdot)$

and $h(\cdot)$, and $i \sim j$ means that the two entries indexed by i and j are "neighbors". For brevity, we often denote $g(\boldsymbol{\mathcal{B}}_i, \boldsymbol{\mathcal{B}}_j)$ as g_{ij} and $h(\boldsymbol{\mathcal{B}}_i) = h_i$ for any i, j. We call (3.7) together with (3.8) as the missing propensity model.

One can interpret the unknown parameter $\boldsymbol{\mathcal{B}}$ as a 1-dimensional feature of each tensor entry. Each neighboring pair of entries i and j contribute to the Hamiltonian via their "co-missingness" $\widetilde{\boldsymbol{\mathcal{W}}}_i\widetilde{\boldsymbol{\mathcal{W}}}_j$ and the interaction of their features $\boldsymbol{\mathcal{B}}_i, \boldsymbol{\mathcal{B}}_j$ through $g(\boldsymbol{\mathcal{B}}_i, \boldsymbol{\mathcal{B}}_j)$. The function $g(\cdot, \cdot)$ describes the tendency of neighboring entries to be observed or missing together. Every entry i also contributes individually to the Hamiltonian via h_i, commonly known as the "external magnetic field" when modeling ferromagnetism. In our context, the function $h(\cdot)$ describes the tendency of each entry to be observed or missing. We provide two concrete examples here to provide the interpretation of the model.

Example 3.2.2 (Independent Bernoulli Model)**.** *Suppose that $g(\cdot, \cdot) = 0$, and let $h(x) = 0.5 \cdot \log f(x)/[1-f(x)]$, where $f(\cdot)$ is an inverse link function (e.g., sigmoid function), then the missing propensity model indicates that for every $s \in [d_1] \times \cdots \times [d_K]$:*

$$\widetilde{\boldsymbol{\mathcal{W}}}_s = \begin{cases} 1, & p = f(\boldsymbol{\mathcal{B}}_s), \\ -1, & p = 1 - f(\boldsymbol{\mathcal{B}}_s), \end{cases} \tag{3.9}$$

and all $\widetilde{\boldsymbol{\mathcal{W}}}_s$ are independent. This independent Bernoulli model nests the previous works that assume missing uniformly at random as a special case.

Example 3.2.3 (Ising Model)**.** *Suppose that $h(\cdot) = 0$, and let $g(x, y) = xy$. Under this scenario, the conditional distribution of $\widetilde{\boldsymbol{\mathcal{W}}}_s$, given all other entries in $\widetilde{\boldsymbol{\mathcal{W}}}$ as $\widetilde{\boldsymbol{\mathcal{W}}}_{-s}$, is:*

$$p(\widetilde{\boldsymbol{\mathcal{W}}}_s = 1 | \boldsymbol{\mathcal{B}}, \widetilde{\boldsymbol{\mathcal{W}}}_{-s}) = \frac{\exp\left[2\boldsymbol{\mathcal{B}}_s \sum_{j \in \mathcal{N}(s)} \widetilde{\boldsymbol{\mathcal{W}}}_j \boldsymbol{\mathcal{B}}_j\right]}{1 + \exp\left[2\boldsymbol{\mathcal{B}}_s \sum_{j \in \mathcal{N}(s)} \widetilde{\boldsymbol{\mathcal{W}}}_j \boldsymbol{\mathcal{B}}_j\right]} = f(\boldsymbol{\mathcal{B}}_s | \sigma_s), \tag{3.10}$$

where $\mathcal{N}(s) = \{j \in [d_1] \times \cdots \times [d_K] | s \sim j\}$, and $f(x|\sigma) = [1 + \exp(-x/\sigma)]^{-1}$ is the sigmoid function with scale parameter σ. This model is similar to the Bernoulli model in (3.9) but has entry-specific scale parameter $\sigma_s = (2\sum_{j \in \mathcal{N}(s)} \widetilde{\boldsymbol{\mathcal{W}}}_j \boldsymbol{\mathcal{B}}_j)^{-1}$ that depends on the missingness and feature of the neighboring entries.

Given the missing propensity model in (3.7) and (3.8), we can compute the p_k according to (3.4) and obtain the conformal weight ω_k as:

$$\omega_k = \frac{p_k}{\sum_{i=1}^{n+1} p_i} = \frac{\exp\left[-2\sum_{s_j \in \mathcal{N}(s_k)} g(\boldsymbol{\mathcal{B}}_{s_k}, \boldsymbol{\mathcal{B}}_{s_j}) \widetilde{\boldsymbol{\mathcal{W}}}_{s_j} - 2h(\boldsymbol{\mathcal{B}}_{s_k})\right]}{\sum_{i=1}^{n+1} \exp\left[-2\sum_{s_j \in \mathcal{N}(s_i)} g(\boldsymbol{\mathcal{B}}_{s_i}, \boldsymbol{\mathcal{B}}_{s_j}) \widetilde{\boldsymbol{\mathcal{W}}}_{s_j} - 2h(\boldsymbol{\mathcal{B}}_{s_i})\right]}, \tag{3.11}$$

with $\widetilde{\mathcal{W}}_s = 1$ for any $s \in \mathbb{S}_{tr} \cup \mathbb{S}_{cal} \cup \{s^*\}$ and $\widetilde{\mathcal{W}}_s = -1$ otherwise. Unfortunately, computing ω_k is still slow in this way because for each $s^* \in \mathbb{S}_{miss}$, we have to temporarily set $\widetilde{\mathcal{W}}_{s^*} = 1$ to compute all the weights. To speed up the computation, we approximate the weight in (3.11) by using the observed binary tensor $\mathcal{W}$ instead, and thus the only difference is that now we have $\mathcal{W}_{s^*} = -1$ for all $s^* \in \mathbb{S}_{miss}$. This approximation makes very little difference since it will only affect those calibration entries in the neighborhood of s^* and the total number of calibration entries is much larger. In the simulation section, we also demonstrate that this approximation has a very negligible impact on the coverage of the conformal interval.

With this approximation, the conformal weight ω_k is now proportional to $(1 - \tilde{p}_{s_k})/\tilde{p}_{s_k}$, where $\tilde{p}_s = p(\widetilde{\mathcal{W}}_s = 1 | [\widetilde{\mathcal{W}}]_{s'} = [\mathcal{W}]_{s'}, \forall s' \neq s)$ is the full conditional probability of entry s being observed given all other entries. The only problem remaining is to estimate the tensor parameter $\mathcal{B}$ using a single sample $\mathcal{W}$, which we will discuss next.

3.3 Estimating Algorithm

In this section, we discuss the details of estimating $\mathcal{B}$ based on a single binary tensor sample $\mathcal{W}$ drawn from the missing propensity model specified by (3.7) and (3.8). More specifically, we attempt to estimate $\mathcal{B}$ using $\mathcal{W}_{\mathbb{S}_{tr}}$, the binary tensor with 1 only in the training set $\mathbb{S}_{tr}$. We describe the estimation framework in Section 3.3.1 and the algorithm in Section 3.3.2.

3.3.1 Low-rank MPLE Framework

Since we only have access to one sample $\mathcal{W}$ and the tensor parameter $\mathcal{B}$ is of the same dimensionality as $\mathcal{W}$, it is infeasible to obtain an estimator $\widehat{\mathcal{B}}$ without imposing additional constraints over $\mathcal{B}$. Similar to previous literature (Wang and Li, 2020; Cai et al., 2022c), we assume that the tensor $\mathcal{B}$ has low tensor rank.

In this chapter, we assume that the tensor $\mathcal{B}$ has a low Tensor-Train (TT) rank (Oseledets, 2011). A low TT-rank tensor $\mathcal{A}$ can be represented by a series of 3-mode TT factor tensors $\mathcal{T}_k \in \mathbb{R}^{r_{k-1} \times d_k \times r_k}, k = 1, \ldots, K, r_0 = r_K = 1$, where for every entry of $\mathcal{A}$, one has:

$$[\mathcal{A}]_{i_1,\ldots,i_K} = [\mathcal{T}_1]_{:i_1:}[\mathcal{T}_2]_{:i_2:} \cdots [\mathcal{T}_K]_{:i_K:}, \tag{3.12}$$

and the right-hand side is a series of matrix multiplication. We say $\mathbf{r} = (r_1, \ldots, r_{K-1})$ is the TT-rank of $\mathcal{A}$ and compactly, we write $\mathcal{A} = [\mathcal{T}_1, \ldots, \mathcal{T}_K]$ and $\mathrm{rank}^{\mathrm{tt}}(\mathcal{A}) = \mathbf{r}$. If one

reshapes the tensor $\boldsymbol{\mathcal{A}}$ into a matrix $\mathbf{A}_k$ of size $\prod_{k' \leq k} d_{k'} \times \prod_{k' > k} d_{k'}$, then the matrix rank of $\mathbf{A}_k$ is r_k. As compared to the more commonly used Tucker rank (Kolda and Bader, 2009), the Tensor-Train rank ensures that the number of parameters representing a low-rank tensor scales linearly with K, the number of modes, making the low TT-rank tensors more efficient for representing high-order tensors.

To ensure the identifiability of TT factors $\boldsymbol{\mathcal{T}}_1, \ldots, \boldsymbol{\mathcal{T}}_K$ in (3.12), it is often required that $\boldsymbol{\mathcal{T}}_1, \ldots, \boldsymbol{\mathcal{T}}_{K-1}$ being *left-orthogonal*. A 3-mode tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ is left-orthogonal if $\mathbf{L}(\boldsymbol{\mathcal{T}})^\top \mathbf{L}(\boldsymbol{\mathcal{T}}) = \mathbf{I}_{d_3 \times d_3}$, where $\mathbf{L}(\cdot) : \mathbb{R}^{d_1 \times d_2 \times d_3} \mapsto \mathbb{R}^{(d_1 d_2) \times d_3}$ is the so-called left-unfolding operator. Finding the representation (3.12) of a low TT-rank tensor with left orthogonality constraint can be achieved by the TT-SVD algorithm (Oseledets, 2011). For completeness, we restate the TT-SVD algorithm in Algorithm 3.1 and denote it as $\text{SVD}^{\text{tt}}_{\mathbf{r}}(\cdot)$.

Algorithm 3.1 Tensor-Train Singular Value Decomposition (TT-SVD)

Input: Tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$, tensor-train rank $\mathbf{r} = (r_1, \ldots, r_{K-1})$.

$\boldsymbol{\mathcal{A}} \leftarrow \boldsymbol{\mathcal{X}}$, $r_0, r_K \leftarrow 1$.
for $k = 1, \ldots, K-1$ **do**
  $\mathbf{A} \leftarrow \text{reshape}[\boldsymbol{\mathcal{A}}, (r_{k-1} d_k, d_{k+1} \cdots d_K)]$. % reshape$(\cdot, \cdot)$ from MATLAB
  Conduct SVD on $\mathbf{A}$ and truncate at rank r_k: $\mathbf{A} \approx \mathbf{U}\mathbf{S}\mathbf{V}^\top$.
  $\boldsymbol{\mathcal{T}}_k \leftarrow \text{reshape}[\mathbf{U}, (r_{k-1}, d_k, r_k)]$.
  $\boldsymbol{\mathcal{A}} \leftarrow \mathbf{S}\mathbf{V}^\top$.
end for
$\boldsymbol{\mathcal{T}}_K \leftarrow \text{reshape}(\boldsymbol{\mathcal{A}}, (r_{K-1}, d_K, 1))$.

Output: Tensor-Train representation $\widehat{\boldsymbol{\mathcal{X}}} = [\boldsymbol{\mathcal{T}}_1, \ldots, \boldsymbol{\mathcal{T}}_K]$ with $\text{rank}^{\text{tt}}(\widehat{\boldsymbol{\mathcal{X}}}) \leq \mathbf{r}$.

Given the assumption that the tensor $\boldsymbol{\mathcal{B}}$ has low TT-rank $\mathbf{r} = (r_1, \ldots, r_{K-1})$, we can re-formulate the MLE of $\boldsymbol{\mathcal{B}}$ as the solution of a low-rank tensor learning problem:

$$\widehat{\boldsymbol{\mathcal{B}}} = \underset{\boldsymbol{\mathcal{B}}: \text{rank}^{\text{tt}}(\boldsymbol{\mathcal{B}}) \leq \mathbf{r}}{\arg\min} \; -\log p(\widetilde{\boldsymbol{\mathcal{W}}} = \boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}} | \boldsymbol{\mathcal{B}}), \tag{3.13}$$

where $\text{rank}^{\text{tt}}(\boldsymbol{\mathcal{B}}) = (r'_1, \ldots, r'_{K-1}) \leq \mathbf{r}$ means that $r'_k \leq r_k$ for any $k = 1, \ldots, K-1$.

However, the likelihood in (3.13) is incorrect since we did not account for the random sampling of the training set, and is also difficult to evaluate its normalizing constant. To circumvent these issues, we consider estimating $\boldsymbol{\mathcal{B}}$ by the maximum pseudo-likelihood estimator (MPLE), which is a common approach for the estimation and inference of Ising model (Ravikumar et al., 2010; Barber and Drton, 2015; Bhattacharya and Mukherjee,

2018). Formally, for each entry i, define $\tilde{p}_i(\boldsymbol{\mathcal{B}})$ as:

$$\begin{aligned}\tilde{p}_i(\boldsymbol{\mathcal{B}}) &= p\left(\widetilde{\boldsymbol{\mathcal{W}}}_i = 1 | [\widetilde{\boldsymbol{\mathcal{W}}}]_s = [\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}]_s, \forall s \neq i, \boldsymbol{\mathcal{B}}\right) \\ &= \frac{\exp\left[2\sum_{j\in\mathcal{N}(i)} g(\boldsymbol{\mathcal{B}}_i, \boldsymbol{\mathcal{B}}_j)[\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}]_j + 2h(\boldsymbol{\mathcal{B}}_i)\right]}{1 + \exp\left[2\sum_{j\in\mathcal{N}(i)} g(\boldsymbol{\mathcal{B}}_i, \boldsymbol{\mathcal{B}}_j)[\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}]_j + 2h(\boldsymbol{\mathcal{B}}_i)\right]}. \end{aligned} \tag{3.14}$$

and we often write it directly as $\tilde{p}_i$. The low-rank MPLE of $\boldsymbol{\mathcal{B}}$ can now be written as:

$$\widehat{\boldsymbol{\mathcal{B}}} = \underset{\boldsymbol{\mathcal{B}}:\text{rank}^{\text{tt}}(\boldsymbol{\mathcal{B}})\leq \mathbf{r}}{\arg\min}\ \ell(\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}|\boldsymbol{\mathcal{B}}) = -\sum_{i:[\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}]_i=1} \log\left(q\tilde{p}_i\right) - \sum_{i:[\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}]_i=-1} \log\left(1-q\tilde{p}_i\right), \tag{3.15}$$

where we $q \in (0,1)$ is the probability of selecting an observed entry into the training set. We discuss the optimization algorithm for solving (3.15) next.

3.3.2 Riemannian Gradient Descent (RGrad) Algorithm

To solve for (3.15), a natural idea is to directly estimate the tensor-train factors $\boldsymbol{\mathcal{T}}_1, \ldots, \boldsymbol{\mathcal{T}}_K$ for $\widehat{\boldsymbol{\mathcal{B}}}$ one at a time, while keeping the others fixed, and iterate until convergence. Such an alternating minimization algorithm has been applied to low-rank binary tensor decomposition (Wang and Li, 2020; Lee and Wang, 2020). However, alternating minimization is computationally inefficient here as each step requires fitting a generalized linear model (GLM) with high-dimensional covariates. Another candidate approach for estimating $\widehat{\boldsymbol{\mathcal{B}}}$ is the projected gradient descent (Chen et al., 2019a), where in each iteration one updates $\boldsymbol{\mathcal{B}}$ along the gradient direction first and then projects it back to the low-rank tensor space with TT-SVD. This is also undesirable since the projection for a high-rank tensor can be very slow.

In this chapter, we propose an optimization technique called Riemannian gradient descent (RGrad), motivated by the fact that rank-$\mathbf{r}$ tensor-train tensors lie on a smooth manifold (Holtz et al., 2012), which we denote as $\mathbb{M}_{\mathbf{r}}$. As compared to the aforementioned methods, RGrad is faster because each step updates $\boldsymbol{\mathcal{B}}$ with a gradient along the tangent space of $\boldsymbol{\mathcal{B}}$, avoiding fitting multiple high-dimensional GLMs. Also, the projection from the tangent space back to the manifold $\mathbb{M}_{\mathbf{r}}$ is faster than the projected gradient descent since the tensors in the tangent space are also low-rank. RGrad has been extensively applied to tensor completion (Kressner et al., 2014; Steinlechner, 2016; Cai et al., 2022d), generalized tensor learning (Cai et al., 2022c) and tensor regression (Luo and Zhang, 2022). The current work, to the best of our knowledge, is the first to apply RGrad to the low TT-rank binary tensor decomposition.

To summarize the RGrad algorithm, we break down the procedures into three steps.

Step I: Compute Vanilla Gradient. We first compute the vanilla gradient $\nabla \ell(\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}|\boldsymbol{\mathcal{B}})$ at the current iterative value $\boldsymbol{\mathcal{B}}$. Formally, the vanilla gradient tensor $\boldsymbol{\mathcal{G}}$ satisfies:

$$[\boldsymbol{\mathcal{G}}]_i = 2 \sum_{j \in \mathcal{N}(i)} (\boldsymbol{\mathcal{V}}_i [\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}]_j + \boldsymbol{\mathcal{V}}_j [\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}]_i)\, g_x(\boldsymbol{\mathcal{B}}_i, \boldsymbol{\mathcal{B}}_j) + 2h'(\boldsymbol{\mathcal{B}}_i)\boldsymbol{\mathcal{V}}_i, \tag{3.16}$$

where $g_x(\cdot,\cdot) = \partial g(\cdot,\cdot)/\partial x$ and $\boldsymbol{\mathcal{V}}_i = (1-\tilde{p}_i)(1-q\tilde{p}_i)^{-1}(q\tilde{p}_i - \mathbb{1}_{\{[\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}]_i=1\}})$, with $\tilde{p}_i$ defined in (3.14).

Step II: Tangent Space Projected Gradient Descent. Suppose that the current iterative value $\boldsymbol{\mathcal{B}}$ has a tensor-train representation $\boldsymbol{\mathcal{B}} = [\boldsymbol{\mathcal{T}}_1, \ldots, \boldsymbol{\mathcal{T}}_K]$. Then any tensor $\boldsymbol{\mathcal{A}}$ within the tangent space $\mathbb{T}$ at $\boldsymbol{\mathcal{B}}$ has an explicit form:

$$\boldsymbol{\mathcal{A}} = \sum_{k=1}^{K} \boldsymbol{\mathcal{C}}_k, \quad \boldsymbol{\mathcal{C}}_k = [\boldsymbol{\mathcal{T}}_1, \ldots, \boldsymbol{\mathcal{T}}_{k-1}, \boldsymbol{\mathcal{Y}}_k, \boldsymbol{\mathcal{T}}_{k+1}, \ldots, \boldsymbol{\mathcal{T}}_K], \tag{3.17}$$

with the constraint that $\mathbf{L}(\boldsymbol{\mathcal{Y}}_k)^\top \mathbf{L}(\boldsymbol{\mathcal{T}}_k) = \mathbf{O}_{r_k \times r_k}$ for all $k < K$, where $\mathbf{O}$ is a zero matrix, and $\boldsymbol{\mathcal{C}}_k$ has the property that $\langle \boldsymbol{\mathcal{C}}_i, \boldsymbol{\mathcal{C}}_j \rangle = 0$ for all $i \neq j$. In this step, one projects the vanilla gradient $\boldsymbol{\mathcal{G}}$ from step I onto $\mathbb{T}$ and obtains the projected gradient $\mathcal{P}_{\mathbb{T}}(\boldsymbol{\mathcal{G}})$. Thanks to the orthogonality of different $\boldsymbol{\mathcal{C}}_k$, the projection problem is solving:

$$\min_{\boldsymbol{\mathcal{Y}}_k : \mathbf{L}(\boldsymbol{\mathcal{Y}}_k)^\top \mathbf{L}(\boldsymbol{\mathcal{T}}_k) = \mathbf{O}_{r_k \times r_k}} \frac{1}{2} \|\boldsymbol{\mathcal{G}} - \boldsymbol{\mathcal{C}}_k\|_{\mathrm{F}}^2, \quad \text{s.t. } \boldsymbol{\mathcal{C}}_k = [\boldsymbol{\mathcal{T}}_1, \ldots, \boldsymbol{\mathcal{T}}_{k-1}, \boldsymbol{\mathcal{Y}}_k, \boldsymbol{\mathcal{T}}_{k+1}, \ldots, \boldsymbol{\mathcal{T}}_K], \tag{3.18}$$

for any $k \leq K-1$ and $\boldsymbol{\mathcal{Y}}_k$ is unconstrained if $k = K$. Solution to (3.18) is:

$$\mathbf{L}(\widehat{\boldsymbol{\mathcal{Y}}}_k) = \left[\mathbf{I}_{r_{k-1}d_k} - \mathbf{L}(\boldsymbol{\mathcal{T}}_k)\mathbf{L}(\boldsymbol{\mathcal{T}}_k)^\top\right] \left(\boldsymbol{\mathcal{B}}^{\leq k-1} \otimes \mathbf{I}_{d_k}\right)^\top \boldsymbol{\mathcal{G}}^{<k>} \left(\boldsymbol{\mathcal{B}}^{\geq k+1}\right)^\top \left[\boldsymbol{\mathcal{B}}^{\geq k+1} \left(\boldsymbol{\mathcal{B}}^{\geq k+1}\right)^\top\right]^{-1}, \tag{3.19}$$

for $k \leq K-1$ and:

$$\mathbf{L}(\widehat{\boldsymbol{\mathcal{Y}}}_K) = \left(\boldsymbol{\mathcal{B}}^{\leq K-1} \otimes \mathbf{I}_{d_K}\right)^\top \boldsymbol{\mathcal{G}}^{<K>}, \tag{3.20}$$

where $\otimes$ is the matrix Kronecker product. In (3.19) and (3.20), $\boldsymbol{\mathcal{G}}^{<k>}$ is the k-mode separation of tensor $\boldsymbol{\mathcal{G}}$, which basically reshapes $\boldsymbol{\mathcal{G}}$ to a matrix of size $\left(\prod_{l \leq k} d_l\right) \times \left(\prod_{l>k} d_l\right)$. Any tensor $\boldsymbol{\mathcal{B}}$ has its k-mode separation as $\boldsymbol{\mathcal{B}}^{<k>} = \boldsymbol{\mathcal{B}}^{\leq k}\boldsymbol{\mathcal{B}}^{\geq k+1}$, where $\boldsymbol{\mathcal{B}}^{\leq k}, \boldsymbol{\mathcal{B}}^{\geq k+1}$ are called the k-th left part and $(k+1)$-th right part. Given that $\boldsymbol{\mathcal{B}} = [\boldsymbol{\mathcal{T}}_1, \ldots, \boldsymbol{\mathcal{T}}_K]$, one can recursively compute $\boldsymbol{\mathcal{B}}^{\leq k}$ as $(\boldsymbol{\mathcal{B}}^{\leq k-1} \otimes \mathbf{I}_{d_k})\mathbf{L}(\boldsymbol{\mathcal{T}}_k)$ and $\boldsymbol{\mathcal{B}}^{\geq k+1}$ as $\mathbf{R}(\boldsymbol{\mathcal{T}}_{k+1})(\mathbf{I}_{d_{k+1}} \otimes \boldsymbol{\mathcal{B}}^{\geq k+2})$ with the convention that $\boldsymbol{\mathcal{B}}^{\leq 0} = \boldsymbol{\mathcal{B}}^{\geq K+1} = 1$, where $\mathbf{R}(\cdot) : \mathbb{R}^{d_1 \times d_2 \times d_3} \mapsto \mathbb{R}^{d_1 \times d_2 d_3}$ is the right-unfolding operator.

After computing $\widehat{\boldsymbol{\mathcal{Y}}}_k$ with (3.19) and (3.20), one ends up with $\widehat{\boldsymbol{\mathcal{C}}}_k = [\boldsymbol{\mathcal{T}}_1, \ldots, \widehat{\boldsymbol{\mathcal{Y}}}_k, \ldots, \boldsymbol{\mathcal{T}}_K]$

and thus the projected gradient $\mathcal{P}_{\mathbb{T}}(\boldsymbol{\mathcal{G}}) = \sum_k \widehat{\boldsymbol{\mathcal{C}}}_k$. This step is completed after one updates $\boldsymbol{\mathcal{B}}$ to $\widetilde{\boldsymbol{\mathcal{B}}} = \boldsymbol{\mathcal{B}} - \eta\mathcal{P}_{\mathbb{T}}(\boldsymbol{\mathcal{G}})$, where η is a constant step size.

Step III: Retraction. As a property of low TT-rank tensors, the updated tensor $\widetilde{\boldsymbol{\mathcal{B}}}$ has its TT-rank upper bounded by $2\mathbf{r}$. To enforce the rank constraint, the last step of RGrad is to retract $\widetilde{\boldsymbol{\mathcal{B}}}$ back to the manifold $\mathbb{M}_{\mathbf{r}}$. We do so by applying TT-SVD to $\widetilde{\boldsymbol{\mathcal{B}}}$: $\boldsymbol{\mathcal{B}}' = \mathrm{SVD}^{\mathrm{tt}}_{\mathbf{r}}(\widetilde{\boldsymbol{\mathcal{B}}})$, and $\boldsymbol{\mathcal{B}}'$ will be the value used for the next iteration.

We summarize the RGrad algorithm in Algorithm 3.2. To provide an initial estimator of $\boldsymbol{\mathcal{B}}$, we apply TT-SVD to a randomly perturbed version of the binary tensor $\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}$, which works quite well empirically. We typically set $\eta = 0.1$ and denote the output of Algorithm 3.2 as $\mathrm{RGrad}(\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}, \mathbf{r})$.

Algorithm 3.2 MPLE of Low-rank Ising Model with Riemannian Gradient Descent

Input: Binary tensor $\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}$, tensor-train rank $\mathbf{r} = (r_1, \ldots, r_{K-1})$, step size η, train-calibration split probability q.

Initialize: let $\boldsymbol{\mathcal{E}} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$ and $\widehat{\boldsymbol{\mathcal{B}}} \leftarrow \mathrm{SVD}^{\mathrm{tt}}_{\mathbf{r}}(\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}} + \boldsymbol{\mathcal{E}}) = [\widehat{\boldsymbol{\mathcal{T}}}_1, \ldots, \widehat{\boldsymbol{\mathcal{T}}}_K]$ by Algorithm 3.1.
for $l = 1, \ldots, l_{\max}$ **do**
  Compute the vanilla gradient $\boldsymbol{\mathcal{G}}$ using (3.16).
  for $k = 1, \ldots, K$ **do**
    Compute $\widehat{\boldsymbol{\mathcal{Y}}}_k$ following (3.19) if $k < K$ and (3.20) if $k = K$.
    $\widehat{\boldsymbol{\mathcal{C}}}_k \leftarrow [\widehat{\boldsymbol{\mathcal{T}}}_1, \ldots, \widehat{\boldsymbol{\mathcal{T}}}_{k-1}, \widehat{\boldsymbol{\mathcal{Y}}}_k, \widehat{\boldsymbol{\mathcal{T}}}_{k+1}, \ldots, \widehat{\boldsymbol{\mathcal{T}}}_K]$.
  end for
  $\mathcal{P}_{\mathbb{T}}(\boldsymbol{\mathcal{G}}) \leftarrow \sum_{k=1}^{K} \widehat{\boldsymbol{\mathcal{C}}}_k$.
  $\widetilde{\boldsymbol{\mathcal{B}}} \leftarrow \widehat{\boldsymbol{\mathcal{B}}} - \eta\mathcal{P}_{\mathbb{T}}(\boldsymbol{\mathcal{G}})$.
  $\widehat{\boldsymbol{\mathcal{B}}} \leftarrow \mathrm{SVD}^{\mathrm{tt}}_{\mathbf{r}}(\widetilde{\boldsymbol{\mathcal{B}}}) = [\widehat{\boldsymbol{\mathcal{T}}}_1, \ldots, \widehat{\boldsymbol{\mathcal{T}}}_K]$ by Algorithm 3.1.
end for

Output: Maximum Pseudo-Likelihood Estimator (MPLE) $\widehat{\boldsymbol{\mathcal{B}}}$ with $\mathrm{rank}^{\mathrm{tt}}(\widehat{\boldsymbol{\mathcal{B}}}) \leq \mathbf{r}$.

By assuming $d_k = O(d), r_k = O(r), \forall k$ and $\max_s |\mathcal{N}(s)| = O(K)$, the computational complexity of RGrad is $O(K(d^K r^2 + dr^3))$ per iteration. See Steinlechner (2016) for more details on the computational complexity of RGrad.

Combining all the discussions in Section 3.2 and 3.3, we summarize the conformalized tensor completion (CTC) algorithm in Algorithm 3.3. We make several remarks for Algorithm 3.3.

Remark 3.3.1 (Fast Entry-wise Quantile Computation)**.** *In the last step of Algorithm 3.3, we compute the empirical $(1-\alpha)$-quantile of the weighted eCDF of the non-conformity score of all calibration data. The for-loop looks slow superficially as one needs to evaluate the quantile for each*

Algorithm 3.3 Conformalized Tensor Completion (CTC)

Input: Data tensor $\boldsymbol{\mathcal{X}}$, tensor-train rank $\mathbf{r}$, train-calibration split probability $q \in (0,1)$, target mis-coverage $\alpha \in (0,1)$, arbitrary tensor completion algorithm $\boldsymbol{\mathcal{A}}$.

$\mathbb{S} \leftarrow \{s \in [d_1] \times \cdots \times [d_K] | \boldsymbol{\mathcal{X}}_s \neq \texttt{NaN}\}$. % indices of entries that are observed

$\boldsymbol{\mathcal{W}} \leftarrow 2 \times \mathbb{1}_{\{s \in \mathbb{S}\}} - 1$.

Randomly partition $\mathbb{S}$ independently into $\mathbb{S}_{tr} \cup \mathbb{S}_{cal}$ with probability q and $1-q$.

$\widehat{\boldsymbol{\mathcal{X}}} \leftarrow \boldsymbol{\mathcal{A}}(\boldsymbol{\mathcal{X}}_{\mathbb{S}_{tr}})$. % $[\boldsymbol{\mathcal{X}}_{\mathbb{S}_{tr}}] = \boldsymbol{\mathcal{X}}_s$ if $s \in \mathbb{S}_{tr}$ and NaN otherwise

$\widehat{\boldsymbol{\mathcal{B}}} \leftarrow \text{RGrad}(\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}, \mathbf{r})$. % RGrad$(\cdot,\cdot)$ is Algorithm 3.2

for $s \in \mathbb{S}_{cal} \cup \mathbb{S}^c$ **do**

  $\tilde{p}_s \leftarrow \left\{1 + \exp\left[-2\sum_{j \in \mathcal{N}(s)} [\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}]_j g(\widehat{\boldsymbol{\mathcal{B}}}_s, \widehat{\boldsymbol{\mathcal{B}}}_j) - 2h(\widehat{\boldsymbol{\mathcal{B}}}_s)\right]\right\}^{-1}$.

  $\omega_s \leftarrow (1-\tilde{p}_s)\tilde{p}_s^{-1}$.

end for

for $s^* \in \mathbb{S}^c$ **do** % See Remark 3.3.1

  Re-normalize $\omega_s, s \in \mathbb{S}_{cal}$ and ω_{s^*} s.t. $\sum_{s \in \mathbb{S}_{cal}} \omega_s + \omega_{s^*} = 1$.

  $\widehat{q}_{s^*} \leftarrow \mathcal{Q}_{1-\alpha}\left(\sum_{s \in \mathbb{S}_{cal}} \omega_s \cdot \delta_{\mathcal{S}(\boldsymbol{\mathcal{X}}_s, \widehat{\boldsymbol{\mathcal{X}}}_s)} + \omega_{s^*} \cdot \delta_{+\infty}\right)$.

end for

Output: $(1-\alpha)$-level conformal interval $C_{1-\alpha,s^*}(\widehat{\boldsymbol{\mathcal{X}}}) \leftarrow \{x \in \mathbb{R} | \mathcal{S}(x, \widehat{\boldsymbol{\mathcal{X}}}_{s^*}) \leq \widehat{q}_{s^*}\}, \forall s^* \in \mathbb{S}^c$.

testing entry s^*. However, $\widehat{q}_{s^*}$ can be computed faster via:

$$\widehat{q}_{s^*} = \begin{cases} +\infty, & \text{if } \omega_{s^*} \geq \alpha, \\ \mathcal{Q}_{\frac{1-\alpha}{1-\omega_{s^*}}}\left(\sum_{s \in \mathbb{S}_{cal}} \frac{\omega_s}{1-\omega_{s^*}} \cdot \delta_{\mathcal{S}(\boldsymbol{\mathcal{X}}_s, \widehat{\boldsymbol{\mathcal{X}}}_s)}\right), & \text{if } \omega_{s^*} < \alpha. \end{cases}$$

which only requires evaluating the quantile of a fixed eCDF shared by all testing entries.

Remark 3.3.2 (Rank Selection)**.** *The implementation of the CTC algorithm requires a proper choice of the tensor-train rank $\mathbf{r}$ for the low-rank Ising model. Typically in low-rank tensor learning literature (Wang and Li, 2020; Cai et al., 2022c), either the Akaike Information Criterion (AIC) (Akaike, 1973) or the Bayesian Information Criterion (BIC) (Schwarz, 1978) is used for the rank selection. Unfortunately, they are not applicable here since we can only compute the pseudo-likelihood. According to literature (Ji and Seymour, 1996; Csiszár and Talata, 2006; Matsuda et al., 2021), one can replace the likelihood in AIC/BIC with pseudo-likelihood and obtain the Pseudo-AIC (P-AIC) and Pseudo-BIC (P-BIC), which are still consistent under some regularity conditions. The P-AIC and P-BIC are defined as:*

$$\text{P-AIC}(r') = 2\ell(\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}} | \widehat{\boldsymbol{\mathcal{B}}}) + 2\left\{\sum_{k=1}^{K-1}\left[d_k r'_{k-1} r'_k - (r'_k)^2\right] + d_K r'_{K-1}\right\}; \tag{3.21}$$

$$\text{P-BIC}(r') = 2\ell(\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}|\widehat{\boldsymbol{\mathcal{B}}}) + \left\{\sum_{k=1}^{K-1}\left[d_k r'_{k-1}r'_k - (r'_k)^2\right] + d_K r'_{K-1}\right\}\log\left(\prod_{k=1}^{K} d_k\right). \tag{3.22}$$

Among all candidate ranks, we select the rank with the smallest P-AIC or P-BIC. In Section B.3.2 of the Appendix, we provide empirical evidence on the consistency of P-AIC and the inconsistency of P-BIC.

3.4 Theoretical Analysis

In this section, we analyze the theoretical property of the CTC algorithm. We mainly focus on the derivation of the error bound of the MPLE estimator $\widehat{\boldsymbol{\mathcal{B}}}$ in Section 3.4.1 and using the derived error bound to further provide a lower bound on the coverage of the conformal intervals in Section 3.4.2.

We assume in this section that $g(x, y) = 0$, i.e. all entries of $\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}$ are observed independently with probability $q(\exp[-2h(\boldsymbol{\mathcal{B}}^*_s)] + 1)^{-1}$. We only analyze the Bernoulli model because of the technical difficulty of handling the Ising model. We leave the theoretical analysis of the Ising model for future works.

3.4.1 MPLE Error Bound

Under the assumption that $g(x, y) = 0$, the MPLE is identical to MLE since the pseudo-likelihood is also the true Bernoulli likelihood. Our main result is in Theorem 3.4.3. To establish the theoretical result, we make several additional assumptions:

Assumption 3.4.1. *$h(\cdot) : \mathbb{R} \mapsto \mathbb{R}$ is a non-decreasing, non-constant twice continuously differentiable function with $h''(\cdot) \geq 0$.*

Assumption 3.4.2. *The MPLE estimator $\widehat{\boldsymbol{\mathcal{B}}}$ and the true tensor parameter $\boldsymbol{\mathcal{B}}^*$ have bounded max-norm: $\|\widehat{\boldsymbol{\mathcal{B}}}\|_\infty, \|\boldsymbol{\mathcal{B}}^*\|_\infty \leq \xi$.*

We define $f(x) = \exp[2h(x)]/(1 + \exp[2h(x)])$ and the following two constants:

$$\alpha_\xi = \sup_{|x|\leq\xi} |2h'(x)|, \quad \gamma_\xi = \inf_{|x|\leq\xi} \min\left\{\left[\frac{f'(x)}{f(x)}\right]^2 - \frac{f''(x)}{f(x)}, \frac{qf''(x)}{1-qf(x)} + \left[\frac{qf'(x)}{1-qf(x)}\right]^2\right\}.$$

To see what these two constants represent, recall that the negative log-likelihood for $\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}$ given $\boldsymbol{\mathcal{B}}$ can be written as the sum of each entry's negative log-likelihood $\ell_i([\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}]_i|\boldsymbol{\mathcal{B}})$,

which is defined as:

$$\ell_i([\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}]_i|\boldsymbol{\mathcal{B}}) = -\left[\left(\frac{[\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}]_i+1}{2}\right)\log qf(\boldsymbol{\mathcal{B}}_i) + \left(\frac{1-[\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}]_i}{2}\right)\log(1-qf(\boldsymbol{\mathcal{B}}_i))\right].$$

It is not difficult to verify that α_ξ upper bounds $|\partial\ell_i(\cdot|\boldsymbol{\mathcal{B}})/\partial\boldsymbol{\mathcal{B}}_i|$ and γ_ξ lower bounds $\partial^2\ell_i(\cdot|\boldsymbol{\mathcal{B}})/\partial\boldsymbol{\mathcal{B}}_i^2$ for all i as long as $\max_s|\boldsymbol{\mathcal{B}}_s| \le \xi$. By excluding the trivial case where $h(\cdot)$ is a constant function, α_ξ is strictly positive. If for all $|x| \le \xi$, we have $1-(1-q)f(x)-f^2(x) > 0$, then we can verify that $\gamma_\xi > 0$ for common choices of $h(\cdot)$, such as the logit model $h(x) = x/2$ or the probit model $h(x) = 2^{-1}\log[\Phi(x)/(1-\Phi(x))]$. For the remainder of the section, we will assume generally that $\gamma_\xi > 0$, which is simply saying that the function $\ell_i(\cdot|\boldsymbol{\mathcal{B}})$ is γ_ξ-strongly convex.

With the aforementioned assumptions and notations, we have the following non-asymptotic bound on $\|\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_{\mathrm{F}}$:

Theorem 3.4.3. *Assume that $g(x,y) = 0$ and assumption 3.4.1 and 3.4.2 hold, and further assume that $\widehat{\boldsymbol{\mathcal{B}}}$ reaches the global minimum of the negative log-likelihood $\ell(\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}|\boldsymbol{\mathcal{B}})$ and the entry-wise negative log-likelihood is γ_ξ-strongly convex with $\gamma_\xi > 0$, then:*

$$\mathrm{P}\left(\frac{1}{\sqrt{d^*}}\|\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_{\mathrm{F}} \le 2C_{K,1}\frac{\alpha_\xi}{\gamma_\xi}\sqrt{\frac{r^*\bar{d}}{d^*}}\right) \ge 1 - \exp\left(-C_1\bar{d}\log K\right), \tag{3.23}$$

where C_1 is an absolute positive constant and $C_{K,1}$ is a positive constant only relates to K.

The proof of the theorem is presented in Appendix B.1.2. We make a remark on Theorem 3.4.3.

Remark 3.4.4. *Under the scenario where $d_1 \asymp \cdots \asymp d_K \asymp O(d)$ and $r_1 \asymp \cdots r_{K-1} \asymp O(r)$, the result in (3.23) can be reduced to:*

$$\mathrm{P}\left(\frac{1}{\sqrt{d^*}}\|\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_{\mathrm{F}} \le 2C_K\frac{\alpha_\xi}{\gamma_\xi}\sqrt{\left(\frac{r}{d}\right)^{K-1}}\right) \ge 1 - \exp\left(-C_1\bar{d}\log K\right).$$

So the estimating error can scale with r/d at a polynomial rate, where lower r/d generally poses an easier binary tensor decomposition problem with lower rooted mean-squared error.

3.4.2 Conformal Inference Coverage Guarantee

In this subsection, we utilize the theoretical result in Theorem 3.4.3 and derive the coverage probability lower bound of the CTC algorithm under the Bernoulli model. The result

will reveal how the estimating error of $\boldsymbol{\mathcal{B}}^*$ propagates into the mis-coverage rate. Our main result is stated in Theorem 3.4.5.

Theorem 3.4.5. *Assume that the same assumptions hold as Theorem 3.4.3 and further denote* $l_\xi = \inf_{|x|\le\xi} \exp[-2h(x)]$, $u_\xi = \sup_{|x|\le\xi} \exp[-2h(x)]$. *The* $(1-\alpha)$*-level conformal interval* $\widehat{C}_{1-\alpha,s}(\widehat{\boldsymbol{\mathcal{X}}})$ *satisfies:*

$$\mathrm{E}\left[\frac{1}{|\mathbb{S}_{miss}|}\sum_{s\in\mathbb{S}_{miss}} \mathbb{1}_{\{\boldsymbol{\mathcal{X}}_s\in\widehat{C}_{1-\alpha,s}(\widehat{\boldsymbol{\mathcal{X}}})\}}\right] \ge 1-\alpha-\frac{2C_{K,1}c_\xi}{(1-c)(1-q)}\sqrt{\frac{r^*\bar{d}}{d^*}} - \exp[-C_1\bar{d}\log K] - \exp\left[-\frac{c^2(1-q)d^*l_\xi}{2}\right], \tag{3.24}$$

for any $0<c<1$, *where* q *is the train-calibration split probability in the CTC algorithm,* $c_\xi = u_\xi\alpha_\xi^2/(\gamma_\xi l_\xi^2)$, *and* $r^*, \bar{d}, d^*$ *are* $\prod_k r_k, \sum_k d_k, \prod_k d_k$ *for* $\boldsymbol{\mathcal{B}}^*$, *respectively.*

We present the proof in Appendix B.1.3 and make a remark on the result of Theorem 3.4.5.

Remark 3.4.6. *Under the scenario where* $d_1 \asymp \cdots \asymp d_K \asymp O(d)$ *and* $r_1 \asymp \cdots r_{K-1} \asymp O(r)$, *the coverage shortfall in* (3.24), *i.e. the difference between the lower bound in* (3.24) *and* $(1-\alpha)$, *can be simplified into:*

$$\frac{c_{K,\xi}}{(1-c)(1-q)}\cdot\sqrt{\left(\frac{r}{d}\right)^{K-1}} + \exp\left[-c_K d\right] + \exp\left[-c'_{K,\xi}c^2(1-q)d^K\right],$$

where $c_{K,\xi}, c'_{K,\xi}$ *are positive constants that only relate to* K *and* ξ. *The first term is of polynomial order with respect to* r/d *while the other two terms are of exponential order with respect to* d, *therefore the first term is the dominating term and the under-coverage of the conformal intervals scale primarily with* $(r/d)^{(K-1)/2}$.

It is a remarkable result that the estimating error, as well as the shortfall of the coverage from the target coverage, increases with $(r^*\bar{d}/d^*)^{1/2}$, where $r^*, \bar{d}, d^*$ are $\prod_k r_k, \sum_k d_k, \prod_k d_k$ for $\boldsymbol{\mathcal{B}}^*$, respectively. If one assumes that $d_k = O(d), r_k = O(r), \forall k$, then the estimation error and coverage shortfall scales with $(r/d)^{(K-1)/2}$. Higher r/d indicates that the data missing pattern is more complex and thus the uncertainty quantification is harder. Although we do not have the theoretical results when $g(x,y) \neq 0$, we show empirically in Section 3.5 that this tendency also holds for the Ising model.

3.5 Simulation Experiments

In this section, we validate the effectiveness of the proposed conformalized tensor completion algorithm via numerical simulations. We consider order-3 cubical tensor of size $d \times d \times d$ and summarize our simulation settings below. Additional details about the simulation setups and results are included in Section B.3 of the Appendix.

3.5.1 Simulation Setup

We simulate the $d \times d \times d$ true tensor parameter $\boldsymbol{\mathcal{B}}^*$ via the Gaussian tensor block model (TBM) (Wang and Zeng, 2019), where $\boldsymbol{\mathcal{B}}^* = \boldsymbol{\mathcal{C}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3 + \boldsymbol{\mathcal{E}}_1$ with $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{r \times r \times r}$ being a core tensor with i.i.d. entries from a Gaussian mixture model: $0.5 \cdot \mathcal{N}(1, 0.5) + 0.5 \cdot \mathcal{N}(-1, 0.5)$, and $\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3 \in \{0,1\}^{d \times r}$ with only a single 1 in each row and $\boldsymbol{\mathcal{E}}_1 \overset{i.i.d.}{\sim} \mathcal{N}(0, 0.2)$. We re-scale the simulated $\boldsymbol{\mathcal{B}}^*$ such that $\|\boldsymbol{\mathcal{B}}^*\|_\infty = 2$. We enforce each column of $\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3$ to have 1 in consecutive rows so that the simulated $\boldsymbol{\mathcal{B}}^*$ demonstrates a noisy "checker box" structure, as illustrated in Figure 3.1(a).

Given the simulated $\boldsymbol{\mathcal{B}}^*$, we then simulate the binary data missingness tensor $\boldsymbol{\mathcal{W}}$ from the Ising model. Throughout this section, we suppose that two tensor entries i and j are neighbors, i.e. $i \sim j$, if and only if their indices differ by 1 in just one mode. Consequently, for 3-way tensors, each non-boundary entry has six neighbors. We simulate $\boldsymbol{\mathcal{W}}$ from the missing propensity model specified by (3.7) and (3.8) with a block-Gibbs sampler and generate samples from a Monte Carlo Markov Chain (MCMC). The MCMC has 4×10^4 iterations with the first 10^4 samples burnt in and we take one sample every other 10^3 iterations to end up with $n = 30$ samples. In Figure 3.1(b), we visualize one simulated $\boldsymbol{\mathcal{W}}$.

Lastly, the data tensor $\boldsymbol{\mathcal{X}}$ is generated from an additive noise model: $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{X}}^* + \boldsymbol{\mathcal{E}}$, which is similar to $\boldsymbol{\mathcal{B}}^*$, with $\boldsymbol{\mathcal{X}}^*$ having a Tucker rank $(3, 3, 3)$. The noiseless tensor $\boldsymbol{\mathcal{X}}^*$ also possesses a "checker box" structure and is contaminated by the noise tensor $\boldsymbol{\mathcal{E}}$, whose distribution depends on the specific simulation setting described later. We re-scale $\boldsymbol{\mathcal{X}}^*$ to have $\|\boldsymbol{\mathcal{X}}^*\|_\infty = 2$ and define the signal-to-noise ratio (SNR) of $\boldsymbol{\mathcal{X}}$ as $\|\boldsymbol{\mathcal{X}}^*\|_\infty / \|\boldsymbol{\mathcal{E}}\|_\infty$ and re-scale $\boldsymbol{\mathcal{E}}$ such that SNR$= 2$. The data tensor $\boldsymbol{\mathcal{X}}$ is then masked by $\boldsymbol{\mathcal{W}}$, as plotted in Figure 3.1(c).

3.5.2 Conformal Prediction Validation

To validate the efficacy of the proposed conformalized tensor completion (CTC) algorithm, we consider the simulation setting with $d \in \{40, 60, 80, 100\}$, $r \in \{3, 5, 7, 9\}$,

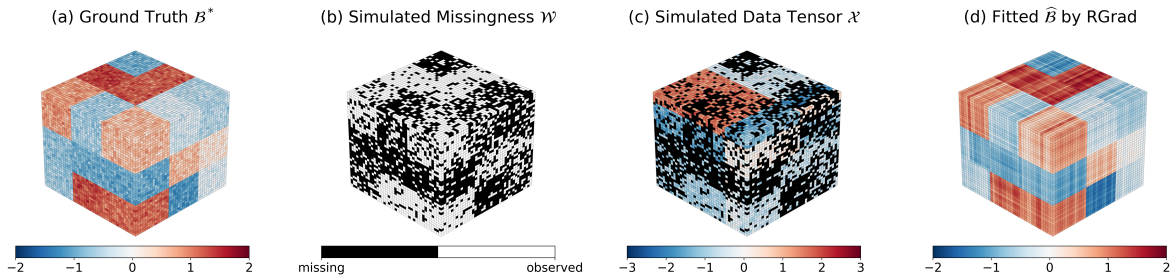


Figure 3.1: Visualizations of key tensors in the simulation setup. (a) Ising model parameter tensor $\boldsymbol{\mathcal{B}}^*$ with $d = 40, r = 3$. (b) Simulated binary tensor $\boldsymbol{\mathcal{W}}$ with $g(x, y) = xy/15, h(x) = x/2$. (c) Simulated data tensor $\boldsymbol{\mathcal{X}}$ masked by $\boldsymbol{\mathcal{W}}$ with $r_0 = 3$, SNR $= 2.0$ and $\boldsymbol{\mathcal{E}}$ having i.i.d. $\mathcal{N}(0, 1)$ entries. (d) Estimated parameter $\widehat{\boldsymbol{\mathcal{B}}}$ from RGrad based on a 70% training set.

$g(x, y) \in \{0, xy/15\}$. The noise tensor $\boldsymbol{\mathcal{E}}$ is simulated based on two different uncertainty regimes: 1) constant noise: $[\boldsymbol{\mathcal{E}}]_s \overset{i.i.d.}{\sim} \mathcal{N}(0, 1)$; 2) adversarial noise: $[\boldsymbol{\mathcal{E}}]_s$ follows independent Gaussian distribution $\mathcal{N}(0, \sigma_s^2)$, with $\sigma_s = [2\exp(\boldsymbol{\mathcal{B}}_s^*)/[1+\exp(\boldsymbol{\mathcal{B}}_s^*)]^{-1}$. The adversarial noise simulates cases where the missing entries have higher uncertainty than the observed entries.

For each simulation scenario, we apply the correctly specified CTC algorithm with P-AIC selected rank and call it **RGrad**. As a benchmark, we also consider two other versions of conformal inference: 1) **unweighted**: the unweighted conformal prediction; 2) **oracle**: the weighted conformal prediction with the true tensor parameter $\boldsymbol{\mathcal{B}}^*$. We conduct simulation over $n = 30$ repetitions, and for each repetition, we randomly split the observed entries into a training and a calibration set with $q = 0.7$ and evaluate the constructed conformal intervals on the missing entries, denoted as $\mathbb{S}_{miss}$. For the tensor completion algorithm, we choose low Tucker rank tensor completion coupled with Riemannian gradient descent (Cai et al., 2022c). We use the absolute residual $\mathcal{S}(y, \widehat{y}) = |y - \widehat{y}|$ as the nonconformity score. To evaluate the conformal intervals, we define the average mis-coverage metric as:

$$\text{Average Mis-coverage \%} = \frac{100}{|\mathbb{Q}|}\sum_{\tau\in\mathbb{Q}}\left|\tau - \frac{1}{|\mathbb{S}_{miss}|}\sum_{s\in\mathbb{S}_{miss}}\mathbb{1}_{\{\boldsymbol{\mathcal{X}}_s\in\widehat{C}_{\tau,s}(\widehat{\boldsymbol{\mathcal{X}}})\}}\right|, \tag{3.25}$$

with $\mathbb{Q} = \{0.80, 0.81, \ldots, 0.98, 0.99\}$. We plot the average mis-coverage with $r = 3$ in Figure 3.2. We also plot the results with $r = 9$ in Section B.3.3 of the Appendix.

According to the results, we find that with constant entry-wise uncertainty, even the unweighted conformal intervals perform decently, but still have more mis-coverage than the

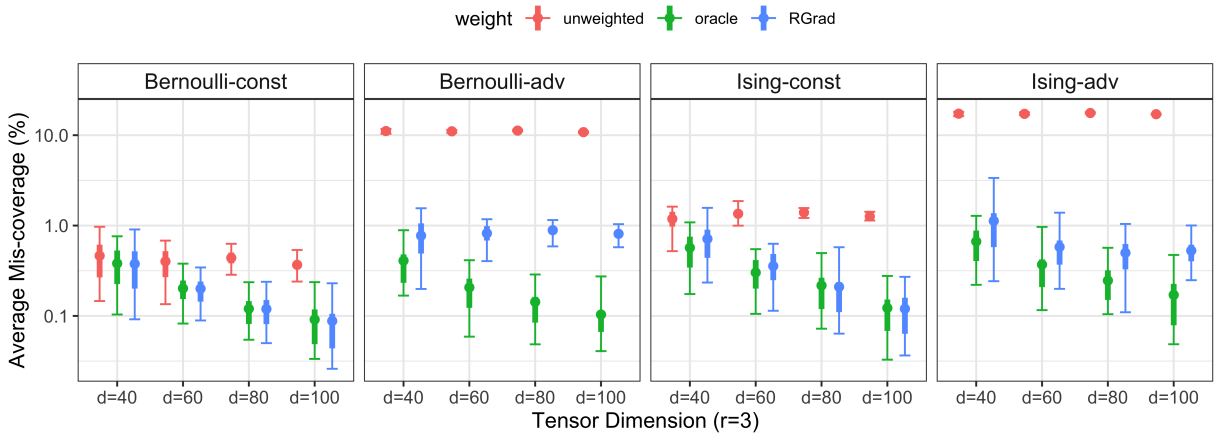


Figure 3.2: The average mis-coverage of three conformal prediction methods with $d \in \{40, 60, 80, 100\}$, $r = 3$ under the Bernoulli and Ising model. Two uncertainty regimes: constant noise (const) and adversarial noise (adv) are considered. Results are based on 30 repetitions, error bars show the $2.5\%, 97.5\%$ quantiles, and the thicker lines show the range of 25% to 75% quantile. The y-axis is plotted in log10-scale.

oracle case. Using our CTC algorithm significantly shrinks the mis-coverage and matches the performance of the oracle case. Under the adversarial noise regime, we observed significant mis-coverage ($> 10\%$) of the unweighted conformal prediction, and using the CTC algorithm provides conformal intervals with $< 1\%$ of mis-coverage, indicating that our method helps in constructing well-calibrated confidence intervals.

The mis-coverage is even worse for the unweighted conformal prediction when missingness is locally dependent based on the Ising model and the CTC algorithm still provides conformal intervals at the target coverage. In Figure B.2 of Section B.3.3 of the Appendix, we further show that the mis-coverage of the unweighted conformal prediction is mainly under-coverage as it cannot account for the increase of uncertainty in the testing set under adversarial noise.

To provide a full landscape on how the conformal intervals based on our CTC algorithm perform under different tensor rank r and tensor dimension d of the underlying parameter $\mathcal{B}^*$, we visualize in Figure 3.3 the empirical coverage of 90% and 95% conformal intervals under different missingness and uncertainty regimes by r/d, i.e. the rank-over-dimension of the tensor $\mathcal{B}^*$, based on our RGrad method. Generally speaking, the higher r/d is, the more difficult it is to estimate the missing propensity of the tensor data and thus the worse the coverage of the conformal intervals, which echoes our theoretical results in Section 3.4. Therefore, we conclude that our proposed method would provide well-calibrated conformal intervals when the underlying missingness model has a low tensor rank relative to

the tensor size (i.e. $r << d$).

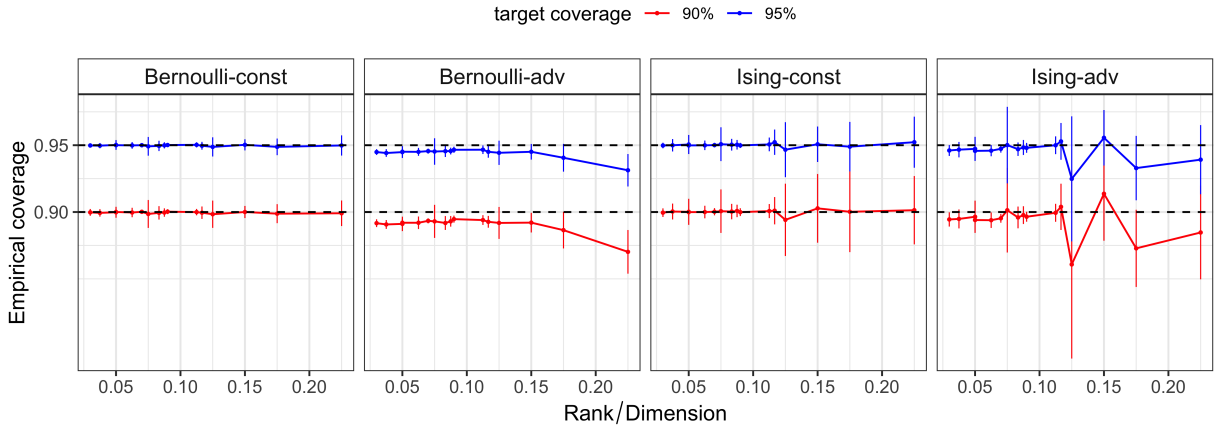


Figure 3.3: RGrad empirical coverage of the 90% and 95% conformal intervals under the Bernoulli and Ising model with two noise regimes. x-axis is the r/d of the tensor parameter $\mathcal{B}^*$. Results are based on $n = 30$ repetitions and error bars are ± 1.96 standard deviations.

In Section B.3.3 of the Appendix, we also compare our RGrad approach with other binary tensor decomposition approaches such as CP and Tucker decomposition for estimating the missing propensity and conducting conformal prediction. We find our method to perform consistently well under all kinds of dependency and uncertainty regimes.

3.6 Data Application to TEC Reconstruction

Our proposed method can account for the locally-dependent data missingness, which is a common data missing pattern for spatial data, therefore we apply our method to a spatio-temporal tensor completion problem in this section as an application. Specifically, we consider the total electron content (TEC) reconstruction problem over the territory of the USA and Canada. The TEC data has severe missing data problems since they can be measured only if the corresponding spatial location has a ground-based receiver. An accurate prediction of the TEC can foretell the impact of space weather on the positioning, navigation, and timing (PNT) service (Wang et al., 2021b; Younas et al., 2022). Existing literature (Pan et al., 2021; Sun et al., 2022a; Wang et al., 2023) focuses on imputation and prediction of the global and regional TEC and lacks data-driven approaches for quantifying the uncertainty of the imputation and we aim at filling in this gap.

In Figure 3.4(a), we plot the TEC distribution over the USA and Canada from the VISTA TEC database (Sun et al., 2023a), as we briefly mentioned in Chapter 2. The VISTA TEC is a pre-imputed version of the Madrigal TEC (MIT Haystack Observatory, 2012), which

has $> 80\%$ of the data missing globally. We use the VISTA TEC as the ground truth and the Madrigal TEC data missingness to mask out entries in the VISTA TEC to simulate data missingness close to what scientists observe in practice.

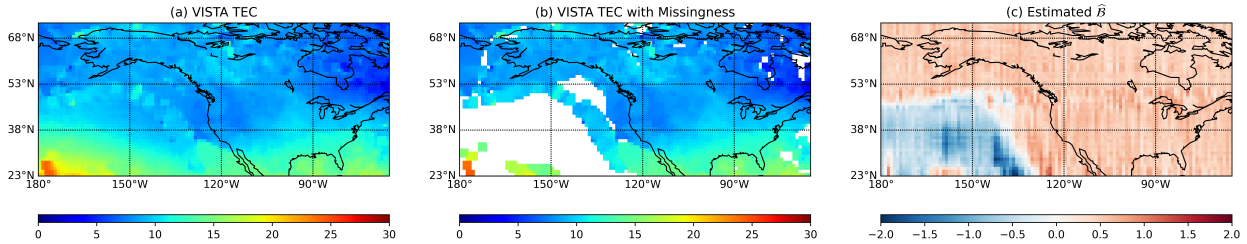


Figure 3.4: (a) The VISTA TEC at 00:02:30 UT, September 1, 2017. (b) The VISTA TEC in (a) with data missingness from the Madrigal TEC. (c) Fitted $\widehat{\boldsymbol{\mathcal{B}}}$ based on the Ising model.

To set up the experiment, we use the first 20 days of data in September 2017, and each day consists of a tensor of size $50 \times 115 \times 96$. We use the first 5 days as a validation set to search for the best $g(\cdot, \cdot)$ function for the Ising model. For each day, we fit the CTC algorithm with a simple tensor completion algorithm based on (3.1) with a Tucker rank at $(3, 3, 3)$ and pick the tensor-train rank $\mathbf{r} = (r, r)$ by P-AIC. Based on Figure 3.3, we know that the Ising model exhibits under-coverage as r/d increases over 0.15, therefore, we select the rank r from $2 \leq r \leq 7$ only. For each day, we consider the Ising model with $g(x, y) = 5xy/4, h(x) = x/2$, the Bernoulli model with $g(x, y) = 0, h(x) = x/2$ and the unweighted conformal prediction for comparison. In Table 3.1, we report the results on the average mis-coverage % and the empirical coverage of 90% and 95% CI.

method	mis-coverage %	90% CI coverage %	95% CI coverage %
unweighted	42.1(6.49)	46.3(6.58)	52.3(7.23)
Bernoulli	23.1(5.34)	64.6(5.97)	76.8(5.03)
Ising	6.01(2.45)	90.0(6.06)	94.2(3.74)

Table 3.1: Mis-coverage % and empirical coverage of CI at 90% and 95% level for the unweighted conformal prediction and weighted conformal prediction with Bernoulli and Ising model for data during Sept 6 to Sept 20, 2017.

In Figure 3.5, we visualize the average 95% CI and its empirical coverage for 20 different bins of TEC values on Sept 6, 2017. It is shown that the data missingness is not uniform across different bins of TEC values and different bins have different distributions of the imputation errors (see how the prediction deviates from the truth), making the unweighted conformal prediction less favorable especially when data missingness is high. These empirical results reveal that by accounting for the heterogeneity and the spatial

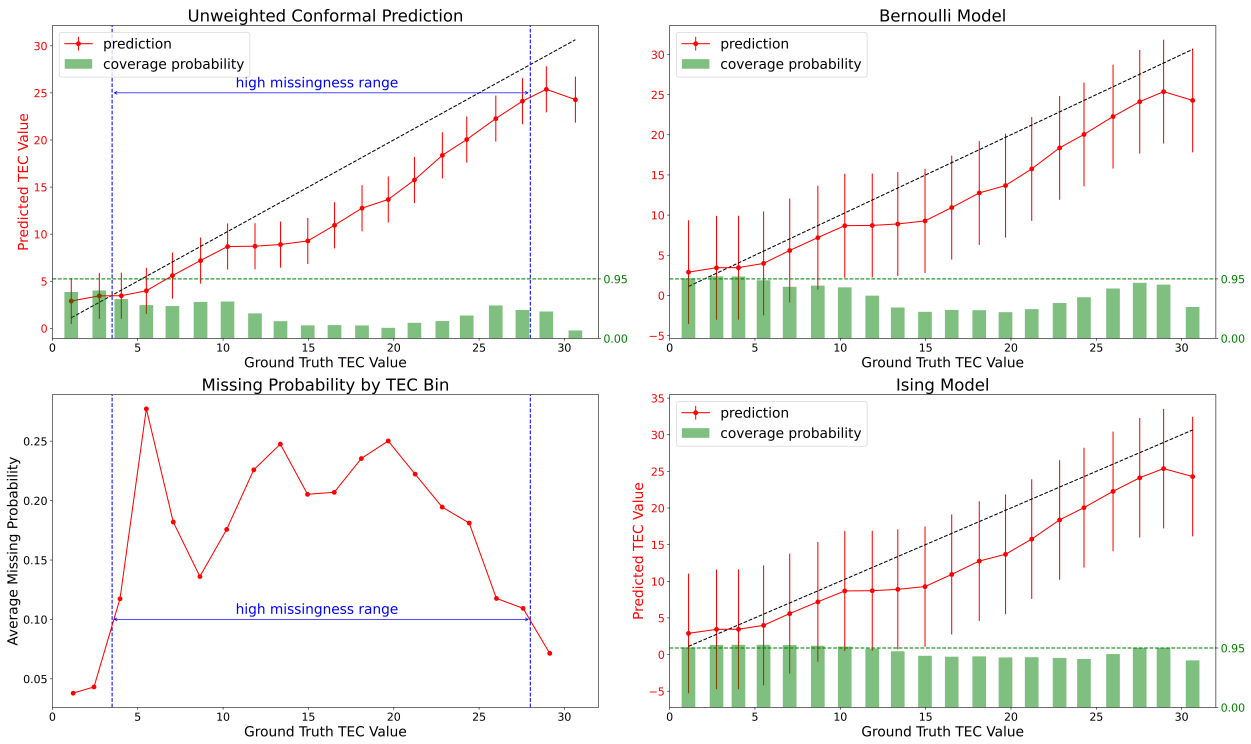


Figure 3.5: All except the lower-left panels show the average 95% conformal intervals and the empirical coverage for 20 different bins of TEC values on Sept 6, 2017. Each bin spans 1.5 TEC units. The lower-left panel shows the missing probability of different bins. A bin is termed "high missingness" if $> 10\%$ of the data is missing.

dependency of data missingness, one can construct well-calibrated confidence intervals using our method.

3.7 Conclusion

In this chapter, we propose a data-driven approach for quantifying the uncertainty of tensor completion. Our method consists of two major steps. We first estimate the missing propensity of each tensor entry using a parameterized Ising model and then plug in the missing propensity estimator to weight each tensor entry and then construct the confidence region with split conformal prediction. We implement the estimation of missing propensity with a computationally efficient Riemannian gradient descent algorithm and validate the resulting conformal intervals with extensive simulation studies and an application to regional TEC reconstruction. There are two limitations of our method. Firstly, we do not have a systematic approach to determine the best specification of the Ising model. Secondly, our Ising model can only account for locally dependent missingness but not

arbitrary missingness. We leave these topics for future research.

CHAPTER 4

Matrix Autoregression with Vector Time-Series Covariates

4.1 Introduction

Matrix-valued time series data have received increasing attention in multiple scientific fields, such as economics (Wang et al., 2019), geophysics (Sun et al., 2022a), and environmental science (Dong et al., 2020), where scientists are interested in modeling the joint dynamics of data observed on a 2-D grid across time. This chapter focuses on the matrix-valued data defined on a 2-D spatial grid that contains the geographical information of the individual observations. As a concrete example, we visualize the global Total Electron Content (TEC) distribution in Figure 4.1. TEC is the density of electrons in the Earth's ionosphere along the vertical pathway connecting a radio transmitter and a ground-based receiver. An accurate prediction of the global TEC is critical since it can foretell the impact of space weather on the positioning, navigation, and timing (PNT) service (Wang et al., 2021b; Younas et al., 2022). Every image in panels (A)-(C) is a 71×73 matrix, distributed on a spatial grid with $2.5°$-latitude-by-$5°$-longitude resolution.

The matrix-valued time series, such as the TEC time series, is often associated with auxiliary vector time series that measures the same object, such as the Earth's ionosphere, from a different data modality. In panel (D) of Figure 4.1, we plot the global SYM-H index, which measures the geomagnetic activity caused by the solar eruptions that can finally impact the global TEC distribution. These non-spatial auxiliary covariates carry additional information related to the matrix time series data dynamics.

In this chapter, we investigate the problem of forecasting future matrix data jointly with the historical matrices and the vector time series covariates. There are two major challenges in this modeling context. From the perspective of building a matrix-variate regression model, we need to integrate the information of predictors with non-uniform modes,

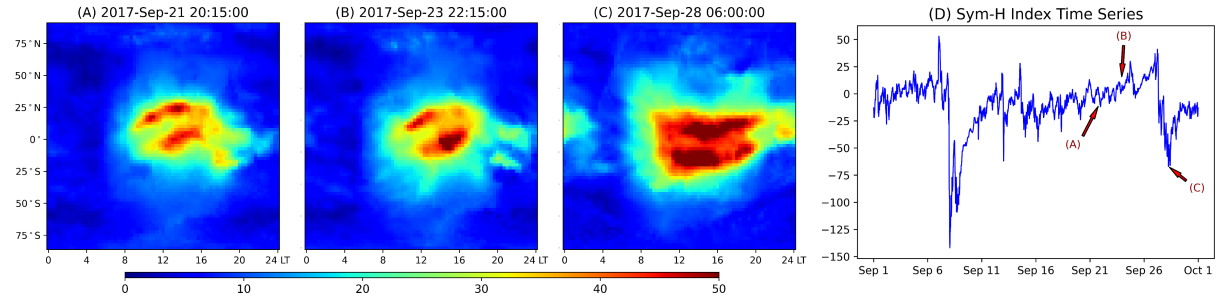


Figure 4.1: An example of matrix time series with auxiliary vector time series. Panels (A)-(C) show the global Total Electron Content (TEC) distribution at three timestamps on the latitude-local-time grid (source: the IGS TEC database (Hernández-Pajares et al., 2009)). Panel (D) plots the auxiliary Sym-H index time series, which measures the impact of solar eruptions on Earth. We highlight the time of panels (A)-(C) in (D) with arrows.

namely both matrices and vectors. Adding the auxiliary vector covariates benefits the prediction and enables domain scientists to understand the interplay between different data modalities but at the same time, it complicates the modeling and the subsequent theoretical analysis. From the perspective of spatio-temporal analysis (Cressie and Wikle, 2015), we need to properly leverage the spatial information of the data and transform the classical spatial statistics framework to accommodate the grid geometry of matrix-valued data. In the remainder of this section, we briefly review the related literature that can shed light on these challenges and then summarize our unique contributions.

A naive yet straightforward prediction model is to vectorize the matrices as vectors and make predictions via the Vector Autoregression (VAR) (Stock and Watson, 2001). In this approach, the auxiliary vector covariates can be incorporated easily by concatenating them with the vectorized matrix predictors. However, vectorizing matrix data leads to the loss of spatial information and also requires a significant amount of parameters given the high dimensionality of the data. To avoid vectorizing the matrix data, scalar-on-tensor regression (Zhou et al., 2013; Guhaniyogi et al., 2017; Li et al., 2018; Papadogeorgou et al., 2021) tackles the problem by using matrix predictors directly. However, these models are built for *scalar* responses while in our setting we are dealing with *matrix* responses. Dividing the matrix response into individual scalar responses and fitting scalar-on-tensor regression still requires a significant number of parameters and more importantly, it fails to take the spatial information of the response into account.

The statistical framework that can incorporate matrices as both predictors and response is the tensor-on-tensor regression (Lock, 2018; Liu et al., 2020; Luo and Zhang, 2022) and more specifically for time series data, the matrix/tensor autoregression (Chen et al., 2021a;

Li and Xiao, 2021; Hsu et al., 2021; Wang et al., 2024). The matrix/tensor predictors are mapped to matrix/tensor responses via multi-linear transformations that greatly reduce the number of parameters. Our work builds on this framework and incorporates the non-spatial vector predictors at the same time.

To incorporate the vector predictor in the same model, we need to map vector predictors to matrix responses. Tensor-on-scalar regression (Rabusseau and Kadri, 2016; Sun and Li, 2017; Li and Zhang, 2017; Guha and Guhaniyogi, 2021) illustrates a way of mapping low-order scalar/vector predictors to high-order matrix/tensor responses via taking the tensor-vector product of the predictor with a high-order tensor coefficient. In this chapter, we take a similar approach and introduce a 3-D tensor coefficient for the vector predictors such that our model can take predictors with non-uniform modes, which is a key distinction of our model compared to existing works.

The other distinction of our model is that our model leverages the spatial information of the matrix response. In our model, a key assumption is that the vector predictor has similar predictive effects on neighboring locations in the matrix response. This is equivalent to saying that the tensor coefficient is spatially smooth. Typically, the estimation of spatially smooth tensor coefficients in such regression models is done via adding a total-variation (TV) penalty (Wang et al., 2017; Shen et al., 2022; Sun et al., 2023b) to the unknown tensor. The TV penalty leads to piecewise smooth estimators with sharp edges and enables feature selections. However, the estimation with the TV penalty requires solving non-convex optimization problems, making the subsequent theoretical analysis difficult. In our model, we utilize a simpler approach by assuming that the tensor coefficients are discrete evaluations of functional parameters from a Reproducing Kernel Hilbert Space (RKHS). Such a kernel method has been widely used in scalar-on-image regressions (Kang et al., 2018) where the regression coefficients of the image predictor are constrained to be spatially smooth.

We facilitate the estimation of the unknown functional parameters with the functional norm penalty. Functional norm penalties have been widely used for estimating smooth functions in classic semi/non-parametric learning in which data variables are either scalar/vector-valued (see Hastie et al., 2009; Gu, 2013; Yuan and Cai, 2010; Cai and Yuan, 2012; Shang and Cheng, 2013, 2015; Cheng and Shang, 2015; Yang et al., 2020a). To the best of our knowledge, the present article is the first to consider functional norm penalty for tensor coefficient estimation in a matrix autoregressive setting.

To encapsulate, this chapter has two major contributions. Firstly, we build a unified matrix autoregression framework for spatio-temporal data that incorporates lower-order scalar/vector time series covariates. Such a framework has strong application motivation where domain scientists are curious about integrating the information of spatial and non-

spatial data for predictions and inference. The framework also bridges regression methodologies with tensor predictors and responses of non-uniform modes, making the theoretical investigation itself an interesting topic. Secondly, we propose to estimate coefficients of the auxiliary covariates, together with the autoregressive coefficients, in a single penalized maximum likelihood estimation (MLE) framework with the RKHS functional norm penalty. The RKHS framework builds spatial continuity into the regression coefficients. We establish the joint asymptotics of the autoregressive coefficients and the functional parameters under fixed/high matrix dimensionality regimes and propose an efficient alternating minimization algorithm for estimation and validate it with extensive simulations and real applications.

The remainder of the chapter is organized as follows. We introduce our model formally in Section 4.2 and provide model interpretations and comparisons in sufficient detail. Section 4.3 introduces the penalized MLE framework and describes an alternating minimization framework for estimation. Large sample properties of the estimators under fixed and high matrix dimensionality are established in Section 4.4. Section 4.5 provides extensive simulation studies for validating the consistency of the estimators, demonstrating BIC-based model selection results, and comparing our method with various competitors. We apply our method to the global TEC data in Section 4.6 and make conclusions in Section 4.7. Technical proofs and additional details of the algorithm and simulations are deferred to Appendix C.

4.2 Model

4.2.1 Notation

We adopt the same notations as defined in Section 1.3. To subscript any tensor/matrix/vector, we use square brackets with subscripts such as $[\boldsymbol{\mathcal{G}}]_{ijd}$, $[\mathbf{z}_t]_d$, $[\mathbf{X}_t]_{ij}$, and we keep the subscript t inside the square bracket to index time. Any fibers and slices of tensor are subscript-ed with colons such as $[\boldsymbol{\mathcal{G}}]_{ij:}$, $[\boldsymbol{\mathcal{G}}]_{::d}$ and thus any row and column of a matrix is denoted as $[\mathbf{X}_t]_{i:}$ and $[\mathbf{X}_t]_{:j}$. If the slices of tensor/matrix are based on the last mode such as $[\boldsymbol{\mathcal{G}}]_{::d}$ and $[\mathbf{X}_t]_{:j}$, we will often omit the colons and write as $[\boldsymbol{\mathcal{G}}]_d$ and $[\mathbf{X}_t]_j$ for brevity.

Following Li and Zhang (2017), the *tensor-vector product* between a tensor $\boldsymbol{\mathcal{G}}$ of size $d_1 \times \cdots \times d_{K+1}$ and a vector $\mathbf{z} \in \mathbb{R}^{d_{K+1}}$, denoted as $\boldsymbol{\mathcal{G}} \bar{\times}_{(K+1)} \mathbf{z}$, or simply $\boldsymbol{\mathcal{G}} \bar{\times} \mathbf{z}$, is a tensor of size $d_1 \times \cdots \times d_K$ with $[\boldsymbol{\mathcal{G}} \bar{\times} \mathbf{z}]_{i_1 \ldots i_K} = \sum_{i_{K+1}} [\boldsymbol{\mathcal{G}}]_{i_1 \ldots i_K i_{K+1}} \cdot [\mathbf{z}]_{i_{K+1}}$. For tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$, we use $\mathbf{X}_{(k)} \in \mathbb{R}^{d_k \times \prod_{m \neq k} d_m}$ to denote its k-mode matricization. The Kronecker product

between matrices is denoted via $\mathbf{A} \otimes \mathbf{B}$ and the trace of a square matrix $\mathbf{A}$ is denoted as $\text{tr}(\mathbf{A})$. We use $\bar{\rho}(\cdot), \underline{\rho}(\cdot), \rho_i(\cdot)$ to denote the maximum, minimum and i^{th} largest eigenvalue of a matrix. We use $diag(\mathbf{C}_1, \ldots, \mathbf{C}_d)$ to denote a block diagonal matrix with $\mathbf{C}_1, \ldots, \mathbf{C}_d$ along the diagonal. More details on the related tensor algebra can be found in Kolda and Bader (2009).

For the matrix time series $\mathbf{X}_t \in \mathbb{R}^{M \times N}$ in our modeling context, we assume that all $S = MN$ grid locations are points on an $M \times N$ grid within the domain $\bar{\mathbb{S}} = [0,1]^2$. The collection of all the spatial locations is denoted as $\mathbb{S}$ and any particular element of $\mathbb{S}$ corresponding to the $(i,j)^{\text{th}}$ entry of the matrix is denoted as s_{ij}. We will often index the $(i,j)^{\text{th}}$ entry of the matrix $\mathbf{X}_t$ with a single index $u = i + (j-1)M$ and thus s_{ij} will be denoted as s_u. We use $[N]$ to denote index set, i.e. $[N] = \{1, 2, \ldots, N\}$. Finally, we use $k(\cdot,\cdot) : \bar{\mathbb{S}} \times \bar{\mathbb{S}} \mapsto \mathbb{R}$ to denote a spatial kernel function and $\mathbb{H}_k$ to denote the corresponding Reproducing Kernel Hilbert Space (RKHS).

4.2.2 Matrix AutoRegression with Auxiliary Covariates (MARAC)

Let $\{\mathbf{X}_t, \mathbf{z}_t\}_{t=1}^T$ be a joint observation of the matrix and the auxiliary vector time series with $\mathbf{X}_t \in \mathbb{R}^{M \times N}, \mathbf{z}_t \in \mathbb{R}^D$. To forecast $\mathbf{X}_t$, we propose our Matrix AutoRegression with Auxiliary Covariates, or MARAC, as:

$$\mathbf{X}_t = \sum_{p=1}^{P} \mathbf{A}_p \mathbf{X}_{t-p} \mathbf{B}_p^\top + \sum_{q=1}^{Q} \boldsymbol{\mathcal{G}}_q \bar{\times} \mathbf{z}_{t-q} + \mathbf{E}_t, \tag{4.1}$$

where $\mathbf{A}_p \in \mathbb{R}^{M \times M}, \mathbf{B}_p \in \mathbb{R}^{N \times N}$ are the autoregressive coefficients for the lag-p matrix predictor and $\boldsymbol{\mathcal{G}}_q \in \mathbb{R}^{M \times N \times D}$ is the tensor coefficient for the lag-q vector predictor, and $\mathbf{E}_t$ is a noise term whose distribution will be specified later. The lag parameters P, Q are hyper-parameters of the model and we often refer to the model (4.1) as MARAC(P,Q).

Based on model (4.1), for the $(i,j)^{\text{th}}$ element of $\mathbf{X}_t$, the MARAC(P,Q) specifies the following model:

$$[\mathbf{X}_t]_{ij} = \sum_{p=1}^{P} \left\langle [\mathbf{A}_p]_{i:}^\top [\mathbf{B}_p]_{j:}, \mathbf{X}_{t-p} \right\rangle + \sum_{q=1}^{Q} [\boldsymbol{\mathcal{G}}_q]_{ij:}^\top \mathbf{z}_{t-q} + [\mathbf{E}_t]_{ij}, \tag{4.2}$$

where each autoregressive term is associated with a rank-1 coefficient matrix determined by the specific rows from $\mathbf{A}_p, \mathbf{B}_p$ and each non-spatial auxiliary covariate is associated with a coefficient vector that is location-specific, i.e. $[\boldsymbol{\mathcal{G}}_q]_{ij:}$. It now becomes more evident from (4.2) that the auxiliary vector covariates enter the model via an elementwise

linear model. The autoregressive term utilizes $\mathbf{A}_p, \mathbf{B}_p$ to transform each lag-p predictor in a bi-linear form. Using such bi-linear transformation greatly reduces the total amount of parameters in that each lagged predictor that required M^2N^2 parameters previously now only requires $M^2 + N^2$ parameters.

For the tensor coefficient $\boldsymbol{\mathcal{G}}_q$, we assume that it is spatially smooth. More specifically, we assume that $[\boldsymbol{\mathcal{G}}_q]_{ijd}$ and $[\boldsymbol{\mathcal{G}}_q]_{uvd}$ are similar if s_{ij}, s_{uv} are spatially close. Formally, we assume that each $[\boldsymbol{\mathcal{G}}_q]_d$, i.e. the coefficient matrix for the d^{th} covariate at lag-q, is a discrete evaluation of a function $g_{q,d}(\cdot) : [0,1]^2 \mapsto \mathbb{R}$ on $\mathbb{S}$. Furthermore, each $g_{q,d}(\cdot)$ comes from an RKHS $\mathbb{H}_k$ endowed with the spatial kernel function $k(\cdot,\cdot)$. The spatial kernel function specifies the spatial smoothness of the functional parameters $g_{q,d}(\cdot)$ and thus the tensor coefficient $\boldsymbol{\mathcal{G}}_q$.

An alternative formulation for $\boldsymbol{\mathcal{G}}_q$ would be a low-rank form (Li and Zhang, 2017). We choose locally smooth over low rank to explicitly model the spatial smoothness of the coefficients and avoid tuning the tensor rank of $\boldsymbol{\mathcal{G}}_q$. We leave the low-rank model for future research and focus on the RKHS framework for the current chapter.

Finally, for the additive noise term $\mathbf{E}_t$, we assume that it is i.i.d. from a multivariate normal distribution with a separable Kronecker-product covariance:

$$\mathbf{vec}(\mathbf{E}_t) \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}_c \otimes \boldsymbol{\Sigma}_r\right), \quad t \in [T], \tag{4.3}$$

where $\boldsymbol{\Sigma}_r \in \mathbb{R}^{M\times M}, \boldsymbol{\Sigma}_c \in \mathbb{R}^{N\times N}$ are the row/column covariance components. Such a Kronecker-product covariance is commonly seen in the covariance models for multi-way data (Hoff, 2011; Fosdick and Hoff, 2014) with the merit of reducing the number of parameters significantly.

Compared to existing models that can only deal with either matrix or vector predictors, our model (4.1) can incorporate predictors with non-uniform modes. If one redefines $\mathbf{E}_t$ in our model as $\sum_{q=1}^{Q} \boldsymbol{\mathcal{G}}_q \bar{\times} \mathbf{z}_{t-q} + \mathbf{E}_t$, i.e. all terms except the autoregressive term, then our model ends up specifying:

$$\begin{aligned}
\mathrm{Cov}(\mathbf{vec}(\mathbf{E}_t), \mathbf{vec}(\mathbf{E}_{t'})) &= \mathbb{1}_{\{t=t'\}} \cdot \boldsymbol{\Sigma}_c \otimes \boldsymbol{\Sigma}_r + \mathbf{F}\mathbf{M}\mathbf{F}^\top, \\
\mathbf{F} = [(\boldsymbol{\mathcal{G}}_1)_{(3)}^\top : \cdots : (\boldsymbol{\mathcal{G}}_Q)_{(3)}^\top], &\quad \mathbf{M} = \left[\mathrm{Cov}(\mathbf{z}_{t-q_1}, \mathbf{z}_{t'-q_2})\right]_{q_1,q_2\in[Q]},
\end{aligned}$$

where $(\boldsymbol{\mathcal{G}}_q)_{(3)}$ is the mode-3 matricization of $\boldsymbol{\mathcal{G}}_q$ and we will use $\mathbf{G}_q$ to denote it for the rest of the chapter. This new formulation reveals how our model differs from other autoregression models with matrix predictors. The covariance of $\mathbf{E}_t, \mathbf{E}_{t'}$ in our model contains a separable covariance matrix $\boldsymbol{\Sigma}_c \otimes \boldsymbol{\Sigma}_r$ that is based on the matrix grid geometry, a locally smooth coefficient matrix $\mathbf{F}$ that captures the local spatial dependency and an auto-

covariance matrix $\mathbf{M}$ that captures the temporal dependency. Consequently, entries of $\mathbf{E}_t$ are more correlated if either they are spatially/temporally close or they share the same row/column index and are thus more flexible for spatial data distributed on a matrix grid.

As a comparison, in the kriging framework (Cressie, 1986), the covariance of $\mathbf{E}_t, \mathbf{E}_{t'}$ is characterized by a spatio-temporal kernel that captures the dependencies among spatial and temporal neighbors. Such kernel method can account for the local dependency but not the spatial dependency based on the matrix grid geometry. In the matrix autoregression model (Chen et al., 2021a), the authors do not consider the local spatial dependencies among entries of $\mathbf{E}_t$ nor the temporal dependency across different t. In Hsu et al. (2021), the matrix autoregression model is generalized to adapt to spatial data via fixed-rank co-kriging (FRC) (Cressie and Johannesson, 2008) with $\mathrm{Cov}(\mathbf{vec}(\mathbf{E}_t), \mathbf{vec}(\mathbf{E}_{t'})) = \mathbb{1}_{\{t=t'\}} \cdot \mathbf{\Sigma}_c \otimes \mathbf{\Sigma}_r + \mathbf{F}\mathbf{M}\mathbf{F}^\top$, where $\mathbf{M}$ is a $k \times k$ coefficient matrix and $\mathbf{F}$ is a pre-specified $MN \times k$ spatial basis matrix. Such a co-kriging framework does not account for the temporal dependency of the noises nor does it consider the auxiliary covariates. Our model generalizes these previous works to allow for temporally dependent noise with both local and grid spatial dependency.

The combination of (4.1) and (4.3) specifies the complete MARAC(P, Q) model. Vectorizing both sides of (4.1) yields the vectorized MARAC(P, Q) model:

$$\mathbf{x}_t = \sum_{p=1}^{P} (\mathbf{B}_p \otimes \mathbf{A}_p)\, \mathbf{x}_{t-p} + \sum_{q=1}^{Q} \mathbf{G}_q^\top \mathbf{z}_{t-q} + \mathbf{e}_t, \quad \mathbf{e}_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(\mathbf{0}, \mathbf{\Sigma}_c \otimes \mathbf{\Sigma}_r\right), \tag{4.4}$$

where $\mathbf{x}_t = \mathbf{vec}(\mathbf{X}_t), \mathbf{e}_t = \mathbf{vec}(\mathbf{E}_t)$, and recall that $\mathbf{G}_q = (\boldsymbol{\mathcal{G}}_q)_{(3)}$. It is now more evident that the Kronecker-product structure of the autoregressive coefficient matrix and the noise covariance matrix greatly reduce the number of parameters, making the regression estimation feasible given finite samples. The spatially smooth structure of $\mathbf{G}_q$ leverages the spatial information of the spatial data. In the next section, we will discuss the estimating algorithm of the model parameters of MARAC.

4.3 Estimating Algorithm

In this section, we discuss the parameter estimation for the MARAC(P, Q) model (4.1). We propose a penalized maximum likelihood estimator (MLE) in Section 4.3.1 for exact parameter estimation. Then in Section 4.3.2, we outline the model selection criterion for selecting the lag hyper-parameters whose consistency will be validated empirically in Section 4.5. In Appendix C.6.2, we further propose an approximation to the penalized MLE

for faster computation by kernel truncation.

4.3.1 Penalized Maximum Likelihood Estimation (MLE)

To estimate the parameters of the MARAC(P, Q) model, which we denote collectively as $\mathbf{\Theta}$, we propose a penalized maximum likelihood estimation (MLE) approach. Following the distribution assumption on $\mathbf{E}_t$ in (4.3), we can write the negative log-likelihood of $\{\mathbf{X}_t\}_{t=1}^T$ with a squared RKHS functional norm penalty, after dropping the constants, as:

$$\mathfrak{L}_\lambda(\mathbf{\Theta}) = -\frac{1}{T}\sum_{t\in[T]} \ell\left(\mathbf{X}_t|\mathbf{X}_{t-1:P}, \mathbf{z}_{t-1:Q}; \mathbf{\Theta}\right) + \frac{\lambda}{2}\sum_{q\in[Q]}\sum_{d\in[D]} \|g_{q,d}\|^2_{\mathbb{H}_k}, \tag{4.5}$$

where $\ell(\cdot)$ is the conditional log-likelihood of $\mathbf{X}_t$:

$$\ell\left(\mathbf{X}_t|\mathbf{X}_{t-1:P}, \mathbf{z}_{t-1:Q}; \mathbf{\Theta}\right) = -\frac{1}{2}\log|\mathbf{\Sigma}_c \otimes \mathbf{\Sigma}_r| - \frac{1}{2}\mathbf{r}_t^\top\left(\mathbf{\Sigma}_c^{-1} \otimes \mathbf{\Sigma}_r^{-1}\right)\mathbf{r}_t, \tag{4.6}$$

and $\mathbf{r}_t = \mathbf{x}_t - \sum_p(\mathbf{B}_p \otimes \mathbf{A}_p)\mathbf{x}_{t-p} - \sum_q \mathbf{G}_q^\top \mathbf{z}_{t-q}$ is the vectorized residual at t. To estimate the parameters, one needs to solve a constrained minimization problem:

$$\min_{\mathbf{\Theta}} \mathfrak{L}_\lambda(\mathbf{\Theta}), \quad \text{s.t. } g_{q,d}(s_{ij}) = [\boldsymbol{\mathcal{G}}_q]_{ijd}, \text{ for all } s_{ij} \in \mathbb{S}. \tag{4.7}$$

We now define the functional norm penalty in (4.5) explicitly and derive a *finite dimensional equivalent* of the optimization problem above. We assume that the spatial kernel function $k(\cdot,\cdot)$ is continuous and square-integrable, thus it has an eigen-decomposition following the Mercer's Theorem (Williams and Rasmussen, 2006):

$$k(s_{ij}, s_{uv}) = \sum_{r=1}^{\infty} \lambda_r \psi_r(s_{ij})\psi_r(s_{uv}), \quad s_{ij}, s_{uv} \in [0,1]^2, \tag{4.8}$$

where $\lambda_1 \geq \lambda_2 \geq \ldots$ is a sequence of non-negative eigenvalues and $\psi_1, \psi_2, \ldots$ is a set of orthonormal eigen-functions on $[0,1]^2$. The functional norm of function g from the RKHS $\mathbb{H}_k$ endowed with kernel $k(\cdot,\cdot)$ is defined as:

$$\|g\|_{\mathbb{H}_k} = \sqrt{\sum_{r=1}^{\infty} \frac{\beta_r^2}{\lambda_r}}, \qquad \text{where } g(\cdot) = \sum_{r=1}^{\infty} \beta_r \psi_r(\cdot), \tag{4.9}$$

following van Zanten and van der Vaart (2008).

Given any $\lambda > 0$ in (4.5), the generalized representer theorem (Schölkopf et al., 2001)

suggests that the solution of the functional parameters, denoted as $\{\widetilde{g}_{q,d}\}_{q=1,d=1}^{Q,D}$, of the minimization problem (4.7), with all other parameters held fixed, is a linear combination of the representers $\{k(\cdot,s)\}_{s\in\mathbb{S}}$ plus a linear combination of the basis functions $\{\phi_1,\ldots,\phi_J\}$ of the null space of $\mathbb{H}_k$, i.e.,

$$\widetilde{g}_{q,d}(\cdot)=\sum_{s\in\mathbb{S}}\gamma_s k(\cdot,s)+\sum_{j=1}^{J}\alpha_j\phi_j(\cdot),\quad \|\phi_j\|_{\mathbb{H}_k}=0, \tag{4.10}$$

where we omit the subscript (q,d) for the coefficient γ_s,α_j for brevity but they are essentially different for each (q,d). We assume that the null space of $\mathbb{H}_k$ contains only the zero function for the remainder of the chapter. As a consequence of (4.10), the minimization problem in (4.7) can be reduced to a finite-dimensional Kernel Ridge Regression (KRR) problem. We summarize the discussion above in the proposition below:

Proposition 4.3.1. *If $\lambda>0$, the constrained minimization problem in (4.7) is equivalent to the following unconstrained kernel ridge regression problem:*

$$\min_{\boldsymbol{\Theta}}\left\{\frac{1}{2}\log|\boldsymbol{\Sigma}_c\otimes\boldsymbol{\Sigma}_r|+\frac{1}{2T}\sum_{t\in[T]}\mathbf{r}_t^\top\left(\boldsymbol{\Sigma}_c^{-1}\otimes\boldsymbol{\Sigma}_r^{-1}\right)\mathbf{r}_t+\frac{\lambda}{2}\sum_{q\in[Q]}tr\left(\boldsymbol{\Gamma}_q^\top\mathbf{K}\boldsymbol{\Gamma}_q\right)\right\}, \tag{4.11}$$

where $\mathbf{r}_t=\mathbf{x}_t-\sum_p(\mathbf{B}_p\otimes\mathbf{A}_p)\mathbf{x}_{t-p}-\sum_q\mathbf{K}\boldsymbol{\Gamma}_q\mathbf{z}_{t-q}$ is the vectorized residual, $\mathbf{K}\in\mathbb{R}^{MN\times MN}$ is the kernel Gram matrix with $[\mathbf{K}]_{u_1u_2}=k(s_{i_1j_1},s_{i_2j_2}),s_{i_lj_l}\in\mathbb{S},u_l=i_l+(j_l-1)M,l=1,2$ and $\boldsymbol{\Gamma}_q\in\mathbb{R}^{MN\times D}$ contains the coefficients of the representers with $[\boldsymbol{\Gamma}_q]_{ud}$ being the coefficient for the u^{th} representer $k(\cdot,s_u)$ and the d^{th} auxiliary covariate at lag q.

We give the proof in Appendix C.1. After introducing the functional norm penalty, the original tensor coefficient is now converted to a linear combination of the representer functions with the relationship that $[\boldsymbol{\mathcal{G}}_q]_{ijd}=\langle[\mathbf{K}]_{u:}^\top,[\boldsymbol{\Gamma}_q]_{:d}\rangle$ where $u=i+(j-1)M$.

We attempt to solve the minimization problem in (4.11) with an alternating minimization algorithm (Attouch et al., 2013) where we update one block of parameters at a time while keeping the others fixed. We update the parameters following the order of: $\mathbf{A}_1\to\mathbf{B}_1\to\cdots\to\mathbf{A}_P\to\mathbf{B}_P\to\boldsymbol{\Gamma}_1\to\cdots\to\boldsymbol{\Gamma}_Q\to\boldsymbol{\Sigma}_r\to\boldsymbol{\Sigma}_c\to\mathbf{A}_1\to\cdots$. We choose the alternating minimization algorithm for its simplicity and efficiency. Each step of the algorithm conducts exact minimization over one block of the parameters, leading to a non-increasing sequence of the objective function, which guarantees the convergence of the algorithm towards a local stationary point.

To solve the optimization problem in (4.11) for $\mathbf{A}_p$ at the $(l+1)^{\text{th}}$ iteration, it suffices to

solve the following least-square problem:

$$\min_{\mathbf{A}_p} \left\{ \sum_{t\in[T]} \text{tr}\left(\widetilde{\mathbf{X}}_t(\mathbf{A}_p)^\top \left(\boldsymbol{\Sigma}_r^{(l)}\right)^{-1} \widetilde{\mathbf{X}}_t(\mathbf{A}_p) \left(\boldsymbol{\Sigma}_c^{(l)}\right)^{-1} \right) \right\}, \tag{4.12}$$

where $\widetilde{\mathbf{X}}_t(\mathbf{A}_p)$ is the residual matrix when predicting $\mathbf{X}_t$:

$$\begin{aligned}\widetilde{\mathbf{X}}_t(\mathbf{A}_p) = \mathbf{X}_t &- \sum_{p'<p} \mathbf{A}_{p'}^{(l+1)} \mathbf{X}_{t-p'} \left(\mathbf{B}_{p'}^{(l+1)}\right)^\top - \sum_{p'>p} \mathbf{A}_{p'}^{(l)} \mathbf{X}_{t-p'} \left(\mathbf{B}_{p'}^{(l)}\right)^\top \\ &- \sum_{q\in[Q]} \boldsymbol{\mathcal{G}}_q^{(l)} \bar{\times} \mathbf{z}_{t-q} - \mathbf{A}_p \mathbf{X}_{t-p} \left(\mathbf{B}_p^{(l)}\right)^\top = \widetilde{\mathbf{X}}_{t,-p} - \mathbf{A}_p \mathbf{X}_{t-p} \left(\mathbf{B}_p^{(l)}\right)^\top,\end{aligned}$$

and we use $\widetilde{\mathbf{X}}_{t,-p}$ to denote the partial residual excluding the term involving $\mathbf{X}_{t-p}$ and use $\boldsymbol{\mathcal{G}}_q^{(l)}$ to denote the tensor coefficient satisfying $[\boldsymbol{\mathcal{G}}_q^{(l)}]_{ijd} = \langle [\mathbf{K}]_{u:}^\top, [\boldsymbol{\Gamma}_q^{(l)}]_{:d} \rangle$, with $u = i + (j-1)M$. The superscript l represents the value at the l^{th} iteration. To simplify the notation, we define $\boldsymbol{\Phi}(\mathbf{A}_t, \mathbf{B}_t, \boldsymbol{\Sigma}) = \sum_t \mathbf{A}_t^\top \boldsymbol{\Sigma}^{-1} \mathbf{B}_t$, where $\boldsymbol{\Sigma}, \mathbf{A}_t, \mathbf{B}_t$ are arbitrary matrices/vectors with conformal matrix sizes and we simply write $\boldsymbol{\Phi}(\mathbf{A}_t, \boldsymbol{\Sigma})$ if $\mathbf{A}_t = \mathbf{B}_t$. Solving (4.12) yields the following updating formula for $\mathbf{A}_p^{(l+1)}$:

$$\mathbf{A}_p^{(l+1)} \leftarrow \boldsymbol{\Phi}\left(\widetilde{\mathbf{X}}_{t,-p}^\top, \mathbf{B}_p^{(l)} \mathbf{X}_{t-p}^\top, \boldsymbol{\Sigma}_c^{(l)}\right) \boldsymbol{\Phi}\left(\mathbf{B}_p^{(l)} \mathbf{X}_{t-p}^\top, \boldsymbol{\Sigma}_c^{(l)}\right)^{-1}. \tag{4.13}$$

Similarly, we have the following updating formula for $\mathbf{B}_p^{(l+1)}$:

$$\mathbf{B}_p^{(l+1)} \leftarrow \boldsymbol{\Phi}\left(\widetilde{\mathbf{X}}_{t,-p}, \mathbf{A}_p^{(l+1)} \mathbf{X}_{t-p}, \boldsymbol{\Sigma}_r^{(l)}\right) \boldsymbol{\Phi}\left(\mathbf{A}_p^{(l+1)} \mathbf{X}_{t-p}, \boldsymbol{\Sigma}_r^{(l)}\right)^{-1}. \tag{4.14}$$

For updating $\boldsymbol{\Gamma}_q$, or its vectorized version $\boldsymbol{\gamma}_q = \text{vec}(\boldsymbol{\Gamma}_q)$, it is required to solve the following kernel ridge regression problem:

$$\min_{\boldsymbol{\gamma}_q} \left\{ \frac{1}{2T} \boldsymbol{\Phi}\left(\widetilde{\mathbf{x}}_{t,-q} - \left(\mathbf{z}_{t-q}^\top \otimes \mathbf{K}\right) \boldsymbol{\gamma}_q, \boldsymbol{\Sigma}^{(l)}\right) + \frac{\lambda}{2} \boldsymbol{\gamma}_q^\top \left(\mathbf{I}_D \otimes \mathbf{K}\right) \boldsymbol{\gamma}_q \right\},$$

where $\boldsymbol{\Sigma}^{(l)} = \boldsymbol{\Sigma}_c^{(l)} \otimes \boldsymbol{\Sigma}_r^{(l)}$ and $\widetilde{\mathbf{x}}_{t,-q}$ is the vectorized partial residual of $\mathbf{X}_t$ by leaving out the lag-q auxiliary predictor, defined in a similar way as $\widetilde{\mathbf{X}}_{t,-p}$. Solving the kernel ridge regression leads to the following updating formula for $\boldsymbol{\gamma}_q^{(l+1)}$:

$$\boldsymbol{\gamma}_q^{(l+1)} \leftarrow \left[\left(\sum_{t\in[T]} \mathbf{z}_{t-q} \mathbf{z}_{t-q}^\top \right) \otimes \mathbf{K} + \lambda T \left(\mathbf{I}_D \otimes \boldsymbol{\Sigma}^{(l)}\right) \right]^{-1} \left[\sum_{t\in[T]} \left(\mathbf{z}_{t-q} \otimes \widetilde{\mathbf{x}}_{t,-q}\right) \right]. \tag{4.15}$$

The step in (4.15) can be slow since one needs to invert a square matrix of size $MND \times MND$. In Appendix C.6.2, we propose an approximation to (4.15) to avoid inverting large matrices.

The updating rule of $\mathbf{\Sigma}_r^{(l+1)}$ and $\mathbf{\Sigma}_c^{(l+1)}$ can be easily derived by taking their derivative in (4.11) and setting it to zero. Specifically, we have:

$$\mathbf{\Sigma}_r^{(l+1)} \leftarrow \frac{1}{NT}\mathbf{\Phi}\left(\widetilde{\mathbf{X}}_t^\top, \mathbf{\Sigma}_c^{(l)}\right), \tag{4.16}$$

$$\mathbf{\Sigma}_c^{(l+1)} \leftarrow \frac{1}{MT}\mathbf{\Phi}\left(\widetilde{\mathbf{X}}_t, \mathbf{\Sigma}_r^{(l+1)}\right), \tag{4.17}$$

where $\widetilde{\mathbf{X}}_t$ is the full residual when predicting $\mathbf{X}_t$.

The algorithm cycles through (4.13), (4.14), (4.15), (4.16) and (4.17) and terminates when $\mathbf{B}_p^{(l)} \otimes \mathbf{A}_p^{(l)}$, $\boldsymbol{\mathcal{G}}_q^{(l)}$, $\mathbf{\Sigma}_c^{(l)} \otimes \mathbf{\Sigma}_r^{(l)}$ have their relative changes between iterations fall under a pre-specified threshold. We make two additional remarks on the algorithm:

Remark 4.3.2. (*Identifiability Constraint*) *The MARAC(P, Q) model specified in (4.1) is scale-unidentifiable in that one can re-scale each pair of $(\mathbf{A}_p, \mathbf{B}_p)$ by a non-zero constant c and obtain $(c \cdot \mathbf{A}_p, c^{-1} \cdot \mathbf{B}_p)$ without changing their Kronecker product. To enforce scale identifiability, we re-scale the algorithm output for each pair of $(\mathbf{A}_p, \mathbf{B}_p)$ such that $\|\mathbf{A}_p\|_{\mathrm{F}} = 1$, sign$(tr(\mathbf{A}_p)) = 1$. The identifiability constraint is enforced before outputting the estimators.*

Remark 4.3.3. (*Convergence of Kronecker Product*) *When dealing with high-dimensional matrices, it is cumbersome to compute the change between $\mathbf{B}_p^{(l)} \otimes \mathbf{A}_p^{(l)}$ and $\mathbf{B}_p^{(l+1)} \otimes \mathbf{A}_p^{(l+1)}$ under the Frobenius norm. An upper bound of $\|\mathbf{B}_p^{(l+1)} \otimes \mathbf{A}_p^{(l+1)} - \mathbf{B}_p^{(l)} \otimes \mathbf{A}_p^{(l)}\|_{\mathrm{F}}$ can be used instead:*

$$\|\mathbf{B}_p^{(l+1)} - \mathbf{B}_p^{(l)}\|_{\mathrm{F}} \cdot \|\mathbf{A}_p^{(l+1)}\|_{\mathrm{F}} + \|\mathbf{B}_p^{(l)}\|_{\mathrm{F}} \cdot \|\mathbf{A}_p^{(l+1)} - \mathbf{A}_p^{(l)}\|_{\mathrm{F}}, \tag{4.18}$$

and a similar bound can be used for the convergence check of $\mathbf{\Sigma}_c^{(l)} \otimes \mathbf{\Sigma}_r^{(l)}$.

4.3.2 Lag Selection

The MARAC(P, Q) model (4.1) has three hyper-parameters: the autoregressive lag P, the auxiliary covariate lag Q, and the RKHS norm penalty weight λ. In practice, λ can be chosen by cross-validation while choosing P and Q requires a more formal model selection criterion. We propose to select P and Q by using information criterion, including the Akaike Information Criterion (AIC) (Akaike, 1973) and the Bayesian Information Criterion (BIC) (Schwarz, 1978). We formally define the AIC and BIC for the MARAC(P, Q) model here and empirically validate their consistency via simulation experiments in Section 4.5.

Let $\widehat{\boldsymbol{\Theta}}$ be the set of the estimated parameters of the MARAC(P, Q) model, and $\mathbf{df}_{P,Q,\lambda}$ be the *effective degrees of the freedom* of the model. We can then define the AIC and the BIC as follows:

$$\mathrm{AIC}(P, Q, \lambda) = -2 \sum_{t \in [T]} \ell(\mathbf{X}_t | \mathbf{X}_{t-1:P}, \mathbf{z}_{t-1:Q}, \widehat{\boldsymbol{\Theta}}) + 2 \cdot \mathbf{df}_{P,Q,\lambda}; \tag{4.19}$$

$$\mathrm{BIC}(P, Q, \lambda) = -2 \sum_{t \in [T]} \ell(\mathbf{X}_t | \mathbf{X}_{t-1:P}, \mathbf{z}_{t-1:Q}, \widehat{\boldsymbol{\Theta}}) + \log(T) \cdot \mathbf{df}_{P,Q,\lambda}. \tag{4.20}$$

To calculate $\mathbf{df}_{P,Q,\lambda}$, we decompose it into the sum of three components: 1) for each pair of the autoregressive coefficient $\widehat{\mathbf{A}}_p, \widehat{\mathbf{B}}_p$, the model has $(M^2 + N^2 - 1)$ degrees of freedom; 2) for the noise covariance $\widehat{\boldsymbol{\Sigma}}_r, \widehat{\boldsymbol{\Sigma}}_c$, the model has $(M^2 + N^2)$ degrees of freedom; and 3) for the auxiliary covariate functional parameters $\widehat{g}_{q,1}, \ldots, \widehat{g}_{q,D}$, inspired by the kernel ridge regression estimator in (4.15), we define the sum of their degrees of freedom as:

$$\mathbf{df}_q(\widehat{g}) = \mathrm{tr}\left\{ \left[\widetilde{\mathbf{K}} + \lambda \left(\mathbf{I}_D \otimes \widehat{\boldsymbol{\Sigma}}_c \otimes \widehat{\boldsymbol{\Sigma}}_r \right) \right]^{-1} \widetilde{\mathbf{K}} \right\},$$

where $\widetilde{\mathbf{K}} = \left(T^{-1} \sum_{t \in [T]} \mathbf{z}_{t-q} \mathbf{z}_{t-q}^\top \right) \otimes \mathbf{K}$. As $\lambda \to 0$, we have $\mathbf{df}_q(\widehat{g}) \to MND$; namely each covariate has MN free parameters, which then reduces to the element-wise linear regression model. Empirically, we find that the BIC is a consistent lag selection criterion for our model.

4.4 Theoretical Analysis

This section presents the theoretical analyses of the MARAC model. We first formulate the condition under which the matrix and vector time series are *jointly stationary*. Under this condition, we then establish the consistency and asymptotic normality of the penalized MLE under *fixed* matrix dimensionality as $T \to \infty$. Finally, we consider the case where the matrix size goes to infinity as $T \to \infty$ and derive the convergence rate of the penalized MLE estimator and also the optimal order of the functional norm penalty tuning parameter λ. Without loss of generality, we assume that the matrix and vector time series have zero means, and we use $S = MN$ to denote the spatial dimensionality of the matrix data. All proofs are deferred to Appendix C.

4.4.1 Stationarity Condition

To facilitate the theoretical analysis, we make an assumption for the vector time series $\mathbf{z}_t$, which greatly simplifies the theoretical analysis.

Assumption 4.4.1. *The D-dimensional auxiliary vector time series $\{\mathbf{z}_t\}_{t=-\infty}^{\infty}$ follows a stationary VAR($\widetilde{Q}$) process:*

$$\mathbf{z}_t = \sum_{\widetilde{q}=1}^{\widetilde{Q}} \mathbf{C}_{\widetilde{q}} \mathbf{z}_{t-\widetilde{q}} + \boldsymbol{\nu}_t, \tag{4.21}$$

where $\mathbf{C}_{\widetilde{q}} \in \mathbb{R}^{D\times D}$ is the lag-$\widetilde{q}$ transition matrix and $\boldsymbol{\nu}_t$ has independent sub-Gaussian entries and is independent of $\mathbf{E}_t$.

Given Assumption 4.4.1, we now derive the condition for $\mathbf{x}_t$ and $\mathbf{z}_t$ to be *jointly stationary*:

Theorem 4.4.2 (MARAC Stationarity Condition)**.** *Assume that Assumption 4.4.1 holds for the auxiliary time series $\{\mathbf{z}_t\}_{t=-\infty}^{\infty}$, and that the matrix time series $\{\mathbf{X}_t\}_{t=-\infty}^{\infty}$ is generated by the MARAC(P, Q) model in (4.1), then $\{\mathbf{X}_t, \mathbf{z}_t\}_{t=-\infty}^{\infty}$ are jointly stationary if and only if for any $y \in \mathbb{C}$ in the complex plane such that $|y| \leq 1$, we have*

$$\det\left[\mathbf{I}_S - \sum_{p=1}^{P} (\mathbf{B}_p \otimes \mathbf{A}_p)\, y^p\right] \neq 0, \quad \det\left[\mathbf{I}_D - \sum_{\widetilde{q}=1}^{\widetilde{Q}} \mathbf{C}_{\widetilde{q}} y^{\widetilde{q}}\right] \neq 0. \tag{4.22}$$

As a special case where $P = \widetilde{Q} = 1$, the stationarity condition in (4.22) is equivalent to $\bar{\rho}(\mathbf{A}_1) \cdot \bar{\rho}(\mathbf{B}_1) < 1$ and $\bar{\rho}(\mathbf{C}_1) < 1$, where $\bar{\rho}(\cdot)$ is the spectral radius of a square matrix. Based on Theorem 4.4.2, the stationarity of the matrix and vector time series relies on the stationarity of the autoregressive coefficients of the MARAC(P, Q) and VAR($\widetilde{Q}$) models and the tensor coefficients $\boldsymbol{\mathcal{G}}_1, \ldots, \boldsymbol{\mathcal{G}}_Q$ do not affect the stationarity.

4.4.2 Finite Spatial Dimension Asymptotics

In this subsection, we establish the consistency and asymptotic normality of the MARAC model estimators under the scenario that M, N are *fixed*. Given a fixed matrix dimensionality, the functional parameters $g_{q,d} \in \mathbb{H}_k$ can only be estimated at $S = MN$ fixed locations, and thus the asymptotic normality result is established for the corresponding tensor coefficient $\widehat{\boldsymbol{\mathcal{G}}}_q$. In Section 4.4.3, we will discuss the *double* asymptotics when both $S, T \to \infty$. For the remainder of the chapter, we denote the true model coefficient with an asterisk superscript, such as $\mathbf{A}_1^*, \mathbf{B}_1^*, \boldsymbol{\mathcal{G}}_1^*$ and $\boldsymbol{\Sigma}^*$.

To start with, we make an assumption on the Gram matrix $\mathbf{K}$:

Assumption 4.4.3. *The minimum eigenvalue of* $\mathbf{K}$ *is bounded by a positive constant* $\underline{c}$*, i.e.* $\underline{\rho}(\mathbf{K}) = \underline{c} > 0$.

As a result of Assumption 4.4.3, every $\boldsymbol{\mathcal{G}}_q^*$ has a unique kernel decomposition: $\mathbf{vec}(\boldsymbol{\mathcal{G}}_q^*) = (\mathbf{I}_D \otimes \mathbf{K})\boldsymbol{\gamma}_q^*$. With this additional assumption, the first theoretical result we establish is the consistency of the covariance matrix estimator $\widehat{\boldsymbol{\Sigma}} = \widehat{\boldsymbol{\Sigma}}_c \otimes \widehat{\boldsymbol{\Sigma}}_r$, which we summarize in Proposition 4.4.4.

Proposition 4.4.4 (Covariance Consistency)**.** *Assume that* $\lambda \to 0$ *as* $T \to \infty$ *and* S *is fixed, and Assumption 4.4.1, 4.4.3 and the stationarity condition in Theorem 4.4.2 hold, , then* $\widehat{\boldsymbol{\Sigma}} \xrightarrow{p.} \boldsymbol{\Sigma}^*$.

Given this result, we can further establish the asymptotic normality of the other model estimators:

Theorem 4.4.5 (Asymptotic Normality)**.** *Assume that the matrix time series* $\{\mathbf{X}_t\}_{t=-\infty}^{\infty}$ *follows the MARAC*(P, Q) *model (4.1) with i.i.d. noise* $\{\mathbf{E}_t\}_{t=-\infty}^{\infty}$ *following (4.3) and Assumption 4.4.1, 4.4.3 and the stationarity condition in Theorem 4.4.2 hold and* $\lambda = o(T^{-1/2})$*. Additionally, assume that* $\underline{\rho}(\mathrm{Var}([\mathbf{vec}(\mathbf{X}_t)^\top, \mathbf{z}_t^\top]^\top)) = \underline{c}' > 0$*. Then suppose* M, N *are fixed and* P, Q *are known and denote* $\mathbf{A}_p, \mathbf{B}_p^\top$ *as* $\boldsymbol{\alpha}_p$ *and* $\boldsymbol{\beta}_p$ *for any* $p \in [P]$*, the penalized MLE of the MARAC*(P, Q) *model is asymptotically normal:*

$$\sqrt{T}\begin{bmatrix} \widehat{\boldsymbol{\beta}}_1 \otimes \widehat{\boldsymbol{\alpha}}_1 - \boldsymbol{\beta}_1^* \otimes \boldsymbol{\alpha}_1^* \\ \vdots \\ \widehat{\boldsymbol{\beta}}_P \otimes \widehat{\boldsymbol{\alpha}}_P - \boldsymbol{\beta}_P^* \otimes \boldsymbol{\alpha}_P^* \\ \mathbf{vec}(\widehat{\boldsymbol{\mathcal{G}}}_1 - \boldsymbol{\mathcal{G}}_1^*) \\ \vdots \\ \mathbf{vec}(\widehat{\boldsymbol{\mathcal{G}}}_Q - \boldsymbol{\mathcal{G}}_Q^*) \end{bmatrix} \xrightarrow{d.} \mathcal{N}\left(\mathbf{0}, \mathbf{V}\boldsymbol{\Xi}\mathbf{V}^\top\right), \tag{4.23}$$

where $\mathbf{V}$ *is:*

$$\mathbf{V} = \begin{bmatrix} diag(\mathbf{V}_1, \ldots, \mathbf{V}_P) & \mathbf{O} \\ \mathbf{O} & \mathbf{I}_{QD} \otimes \mathbf{K} \end{bmatrix}, \quad \mathbf{V}_p = [\boldsymbol{\beta}_p^* \otimes \mathbf{I}_{M^2}, \mathbf{I}_{N^2} \otimes \boldsymbol{\alpha}_p^*],$$

and $\boldsymbol{\Xi} = \mathbf{H}^{-1}\mathrm{E}\left[\mathbf{W}_t^\top(\boldsymbol{\Sigma}^*)^{-1}\mathbf{W}_t\right]\mathbf{H}^{-1}$*, and* $\mathbf{W}_t$ *is defined as:*

$$\mathbf{W}_t = \left[\mathbf{W}_{0,t} \otimes \mathbf{I}_M, \mathbf{I}_N \otimes \mathbf{W}_{1,t}, [\mathbf{z}_{t-1}^\top, \ldots, \mathbf{z}_{t-Q}^\top] \otimes \mathbf{K}\right],$$

where $\mathbf{W}_{0,t} = [\mathbf{B}_1^* \mathbf{X}_{t-1}^\top, \ldots, \mathbf{B}_P^* \mathbf{X}_{t-P}^\top]$, $\mathbf{W}_{1,t} = [\mathbf{A}_1^* \mathbf{X}_{t-1}, \ldots, \mathbf{A}_P^* \mathbf{X}_{t-P}]$, *and:*

$$\mathbf{H} = \mathrm{E}\left[\mathbf{W}_t^\top (\boldsymbol{\Sigma}^*)^{-1} \mathbf{W}_t\right] + \boldsymbol{\zeta}\boldsymbol{\zeta}^\top, \quad \boldsymbol{\zeta}^\top = \left[(\boldsymbol{\alpha}_1^*)^\top, \cdots, (\boldsymbol{\alpha}_P^*)^\top, \mathbf{0}^\top\right].$$

The asymptotic distribution (4.23) indicates that all parameters are $\sqrt{T}$-consistent. Under a fixed matrix dimensionality S, the functional parameters $g_{q,d} \in \mathbb{H}_k$ are estimated only at fixed locations. Hence, the convergence is at a parametric rate just like the autoregressive coefficient.

4.4.3 High Spatial Dimension Asymptotics

The previous section presents the asymptotic normality of the MARAC estimators under a *fixed* matrix dimensionality S. In this section, we relax this assumption and establish the convergence rate of the MARAC estimators when $S, T \to \infty$. For technical reasons, we assume that all entries of $\mathbf{E}_t$ are i.i.d. normally-distributed random variables following $\mathcal{N}(0, \sigma^2)$.

To establish the convergence rate of the MARAC estimators when $S, T \to \infty$, we need to make several additional assumptions.

Assumption 4.4.6. *The spatial locations of the rows and columns of* $\mathbf{X}_t$ *are sampled independently from a uniform distribution on* $[0, 1]$.

Assumption 4.4.7. *The spatial kernel function* $k(\cdot, \cdot)$ *can be decomposed into the product of a row kernel* $k_1(\cdot, \cdot)$ *and a column kernel* $k_2(\cdot, \cdot)$ *that satisfies* $k((u, v), (s, t)) = k_1(u, s)k_2(v, t)$. *Both* k_1, k_2 *have their eigenvalues decaying at a polynomial rate:* $\lambda_j(k_1) \asymp j^{-r_0}, \lambda_j(k_2) \asymp j^{-r_0}$ *with* $r_0 \in (1/2, 2)$.

Assumption 4.4.7 elicits a simple eigen-spectrum characterization of the spatial kernel $k(\cdot, \cdot)$, whose eigenvalue can be written as $\lambda_i(k_1)\lambda_j(k_2)$. Also, the Gram matrix $\mathbf{K}$ is separable, i.e. $\mathbf{K} = \mathbf{K}_2 \otimes \mathbf{K}_1$ and all eigenvalues of $\mathbf{K}$ have the form: $\rho_i(\mathbf{K}_1)\rho_j(\mathbf{K}_2)$, where $\mathbf{K}_1 \in \mathbb{R}^{M \times M}, \mathbf{K}_2 \in \mathbb{R}^{N \times N}$ are the Gram matrix for the kernel k_1, k_2, respectively.

Under Assumption 4.4.6, we further have $\rho_i(\mathbf{K}_1) \to M\lambda_i(k_1)$ and $\rho_j(\mathbf{K}_2) \to N\lambda_j(k_2)$, as $M, N \to \infty$. Combined with Assumption 4.4.7, we can characterize the eigenvalues of $\mathbf{K}$ as $S(ij)^{-r_0}$. We refer our readers to Koltchinskii and Giné (2000); Braun (2006) for more references about the eigen-analysis of the kernel Gram matrix. One can generalize Assumption 4.4.6 to non-uniform sampling, but here, we stick to this simpler setting.

In Assumption 4.4.7, we assume the kernel separability to accommodate the grid structure of the spatial locations. We do not restrict r_0 to be an integer but just a parameter that

characterizes the smoothness of the functional parameters. With these assumptions, we are now ready to present the main result in Theorem 4.4.8.

Theorem 4.4.8 (Asymptotics for High-Dimensional MARAC)**.** *Assume that Assumptions 4.4.1, 4.4.6 and 4.4.7 hold and $\mathbf{X}_t$ is generated by the MARAC(P, Q) model (4.1) with $\mathbf{E}_t$ having i.i.d. $\mathcal{N}(0, \sigma^2)$ entries. Then as $S, T \to \infty$ (D is fixed) and $S \log S/T \to 0$, and under the additional assumptions that:*

1. *$M = O(\sqrt{S}), N = O(\sqrt{S})$;*
2. *$\gamma_S := \lambda/S \to 0$ and $\gamma_S \cdot S^{r_0} \to C_1$ as $S \to \infty$, with $0 < C_1 \leq \infty$;*
3. *$\underline{\rho}(\mathbf{\Sigma}^*_{\mathbf{x},\mathbf{x}} - (\mathbf{\Sigma}^*_{\mathbf{z},\mathbf{x}})^\top (\mathbf{\Sigma}^*_{\mathbf{z},\mathbf{z}})^{-1} \mathbf{\Sigma}_{\mathbf{z},\mathbf{x}}) = c_{0,S} > 0$ as $S, T \to \infty$, where $\mathbf{\Sigma}^*_{\mathbf{x},\mathbf{x}}, \mathbf{\Sigma}^*_{\mathbf{z},\mathbf{z}}, \mathbf{\Sigma}^*_{\mathbf{z},\mathbf{x}}$ are $\mathrm{Var}(\mathbf{x}_t)$, $\mathrm{Var}(\mathbf{z}_t)$ and $\mathrm{Cov}(\mathbf{z}_t, \mathbf{x}_t)$, respectively. $c_{0,S}$ is a constant that only relates to S;*
4. *For any S, we have $0 < \underline{\rho}(\mathbf{K}) < \bar{\rho}(\mathbf{K}) \leq C_0$, where C_0 is a finite constant,*

then we have:

$$\frac{1}{\sqrt{P}S} \sqrt{\sum_{p=1}^{P} \left\| \widehat{\mathbf{B}}_p \otimes \widehat{\mathbf{A}}_p - \mathbf{B}^*_p \otimes \mathbf{A}^*_p \right\|_{\mathrm{F}}^2} \lesssim O_P\left(\sqrt{\frac{C_g \cdot \gamma_S}{c_{0,S} \cdot S}} \right) + O_P\left(\sqrt{\frac{D}{c_{0,S} \cdot TS}} \right), \tag{4.24}$$

where $C_g = \sum_{q=1}^{Q} \sum_{d=1}^{D} \|g_{q,d}\|^2_{\mathbb{H}_k}$. Furthermore, we also have:

$$\begin{aligned} &\sqrt{(TS)^{-1} \sum_{t=1}^{T} \left\| \sum_{q=1}^{Q} \left(\widehat{\boldsymbol{\mathcal{G}}}_q - \boldsymbol{\mathcal{G}}^*_q \right) \bar{\times} \mathbf{z}_{t-q} \right\|_{\mathrm{F}}^2} \\ &\lesssim O_P\left(\frac{\sqrt{\gamma_S^{-1/2r_0}}}{\sqrt{T}\sqrt[4]{S}} \right) + O_P(\sqrt{\gamma_S}) + O_P\left(\frac{1}{\sqrt{S}} \right) + O_P\left(\frac{\sqrt{\gamma_S^{-1}}}{\sqrt{TS}} \right). \end{aligned} \tag{4.25}$$

In Theorem 4.4.8, (4.24) gives the error bound of the autoregressive coefficients and (4.25) gives the error bound of the prediction made by the auxiliary time series, which contains the functional parameter estimators. As a special case of (4.24) where $\gamma_S = 0$ and S is fixed, the convergence rate for the autoregressive coefficients is $O_P(T^{-1/2})$, which reproduces the result in Theorem 4.4.5. For the discussion below, we use AR_{err} and AC_{err} as acronyms for the quantity on the left-hand side of (4.24) and (4.25).

Remark 4.4.9 (Optimal Choice of λ and Phase Transition)**.** *According to our proof, the error bound (4.25) can be decomposed into the sum of:*

- *nonparametric error:* $O_P\left(\sqrt{\frac{\gamma_S^{-1/2r_0}}{T\sqrt{S}}}\right) + O_P(\sqrt{\gamma_S})$,
- *autoregressive error:* $O_P\left(\sqrt{\gamma_S}\right) + O_P\left(S^{-1/2}\right) + O_P\left(T^{-1/2}\right) + O_P\left(\sqrt{\frac{\gamma_S^{-1}}{TS}}\right)$,

where the autoregressive error stems from the estimation error of $\widehat{\mathbf{B}}_p \otimes \widehat{\mathbf{A}}_p$. *The nonparametric error resembles the result of nonparametric regression with RKHS norm penalty (Cui et al., 2018), where if the number of data points is* n *and penalty tuning parameter is* λ, *then the nonparametric error is bounded by* $O_P(\sqrt{\lambda^{-1/2r_0}/n}) + O_P(\sqrt{\lambda})$ *with an optimal* $\lambda \asymp n^{-2r_0/(2r_0+1)}$. *In our model, if there is no autoregressive error, the optimal tuning parameter satisfies* $\gamma_S \asymp (T\sqrt{S})^{-2r_0/(2r_0+1)}$. *The number of data points in our case is* TS, *and we are short of* $\sqrt{S}$ *in the optimal tuning parameter due to Assumption 4.4.7, where the eigenvalues of* $\mathbf{K}$, *ordered as* $\rho_1(\mathbf{K}) \geq \cdots \rho_i(\mathbf{K}) \geq \cdots$, *decay slower than* i^{-2r_0}. *This is a special result for matrix-shaped data. It is also noteworthy that under the condition that* $S \log S/T \to 0$, *the autoregressive error dominates the nonparametric error.*

To simplify the discussion of the optimal order of γ_S, *we assume that* $S = T^c$, *where* $c < 1$ *is a constant. Under this condition, when* $P, Q \geq 1$, *the optimal tuning parameter* $\gamma_S = \lambda/S$ *shows an interesting phase transition phenomenon under different spatial smoothness* r_0 *and matrix dimensionality* $c = \log_T S$, *which we summarize in Table 4.1.*

r_0	$\log_T S$	Optimal γ_S	Estimator Error
$[1, 2)$	$[\frac{1}{2r_0-1}, 1)$	$O((TS)^{-\frac{1}{2}})$	$\mathrm{AR}_{err} = O_P(T^{-\frac{1}{4}}S^{-\frac{3}{4}})$ $\mathrm{AC}_{err} = O_P(S^{-\frac{1}{2}})$
$[1, 2)$	$(0, \frac{1}{2r_0-1})$	$O(S^{-r_0})$	$\mathrm{AR}_{err} = O_P(S^{-\frac{r_0+1}{2}})$ $\mathrm{AC}_{err} = O_P(S^{-\frac{1}{2}})$
$(\frac{1}{2}, 1)$	$[2r_0 - 1, 1)$	$O(S^{-r_0(2r_0-1)})$	$\mathrm{AR}_{err} = O_P(S^{-\frac{r_0(2r_0-1)+1}{2}})$ $\mathrm{AC}_{err} = O_P(S^{-\frac{1}{2}})$
$(\frac{1}{2}, 1)$	$(0, 2r_0 - 1)$	$O((T\sqrt{S})^{-\frac{2r_0}{2r_0+1}})$	$\mathrm{AR}_{err} = O_P((TS)^{-\frac{1}{2}}) + O_P((T\sqrt{S})^{-\frac{r_0}{2r_0+1}}S^{-\frac{1}{2}})$ $\mathrm{AC}_{err} = O_P(S^{-\frac{1}{2}}) + O_P((T\sqrt{S})^{-\frac{r_0}{2r_0+1}})$

Table 4.1: Summary of optimal tuning parameter γ_S and estimators error following (4.24) and (4.25), under the assumption that $c_{0,S} \geq c_0 > 0$, for all S and $S = T^c$ for some constant $0 < c < 1$ such that $S \log S/T \to 0$. AR_{err} and AC_{err} are the quantity on the left-hand side of (4.24) and (4.25).

Based on the results in Table 4.1, the faster S *grows with respect to* T, *the smaller the optimal tuning parameter* γ_S *is. This is an intuitive result since when one has more spatial locations, the observations are denser, and thus less smoothing is needed. Furthermore, we achieve an optimal tuning order of* γ_S *that is close to the classic nonparametric optimal rate at* $(TS)^{-2r_0/(2r_0+1)}$ *only under the regime where* $1/2 < r_0 < 1$ *and* $\log_T S < 2r_0 - 1$. *This regime specifies the scenario*

where the functional parameter is relatively unsmooth and the spatial dimensionality grows slowly with respect to T. Only under this regime will the discrepancy between the nonparametric error and the autoregressive error remain small, leading to an optimal tuning parameter close to the result of nonparametric regression.

In (4.24), the constant $c_{0,S}$ appears in the error bound of the autoregressive term. This constant characterizes the spatial correlation of the matrix time series $\mathbf{X}_t$, conditioning on the auxiliary vector time series $\mathbf{z}_t$ and can vary across different assumptions made on the covariances of $\mathbf{E}_t$ and $\boldsymbol{\nu}_t$. In Table 4.1, we assume that $c_{0,S} \geq c_0 > 0$ for some universal constant c_0. Unfortunately, in practice, it is common to have $c_{0,S} \to 0$ as $S \to \infty$, which makes the autoregressive coefficient converge at a slower rate but does not affect the functional parameter convergence. We leave the constant $c_{0,S}$ here in (4.24) to give a general result and leave the characterization of $c_{0,S}$ under specific assumptions for future works.

4.5 Simulation Experiments

4.5.1 Consistency and Convergence Rate

In this section, we validate the consistency and convergence rate of the MARAC estimators. We consider a simple setup with $P = Q = 1$ and $D = 3$ and simulate the autoregressive coefficients $\mathbf{A}_1^*, \mathbf{B}_1^*$ such that they satisfy the stationarity condition in Theorem 4.4.2. We specify both $\mathbf{A}_1^*, \mathbf{B}_1^*$ and $\boldsymbol{\Sigma}_r^*, \boldsymbol{\Sigma}_c^*$ to have symmetric banded structures. To simulate g_1, g_2, g_3 (we drop the lag subscript) from the RKHS $\mathbb{H}_k$, we choose $k(\cdot, \cdot)$ to be the Lebedev kernel (Kennedy et al., 2013) and generate g_1, g_2, g_3 randomly from Gaussian processes with the Lebedev kernel as the covariance kernel. Finally, we simulate the auxiliary vector time series $\mathbf{z}_t \in \mathbb{R}^3$ from a VAR(1) process. We include more details and visualizations of the simulation setups in Appendix C.6.

The evaluation metric is the rooted mean squared error (RMSE), defined as $\mathrm{RMSE}(\widehat{\boldsymbol{\Theta}}) = \|\widehat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}^*\|_{\mathrm{F}} / \sqrt{d(\boldsymbol{\Theta}^*)}$, where $d(\boldsymbol{\Theta}^*)$ is the number of elements in $\boldsymbol{\Theta}^*$. We consider $\boldsymbol{\Theta} \in \{\mathbf{B}_1 \otimes \mathbf{A}_1, \boldsymbol{\Sigma}_c \otimes \boldsymbol{\Sigma}_r, \boldsymbol{\mathcal{G}}_1, \boldsymbol{\mathcal{G}}_2, \boldsymbol{\mathcal{G}}_3\}$ and we report the average RMSE for $\boldsymbol{\mathcal{G}}_1, \boldsymbol{\mathcal{G}}_2, \boldsymbol{\mathcal{G}}_3$. The dataset is configured with $M \in \{5, 10, 20, 40\}$ and $N = M$. For each M, we train the MARAC model with $P = Q = 1$ over $T_{\text{train}} \in \{1, 5, 10, 20, 40, 80, 160\} \times 10^2$ frames of the matrix time series and choose the tuning parameter λ based on the prediction RMSE over a held-out validation set with $T_{\text{val}} = T_{\text{train}}/2$ and we validate the prediction performance over a $5,000$-frame testing set. We simulate a sufficiently long time series and choose the training set starting from the first frame and the validation set right after the training set.

The testing set is always fixed as the last $5,000$ frames of the time series. All results are reported with 20 repetitions in Figure 4.2.

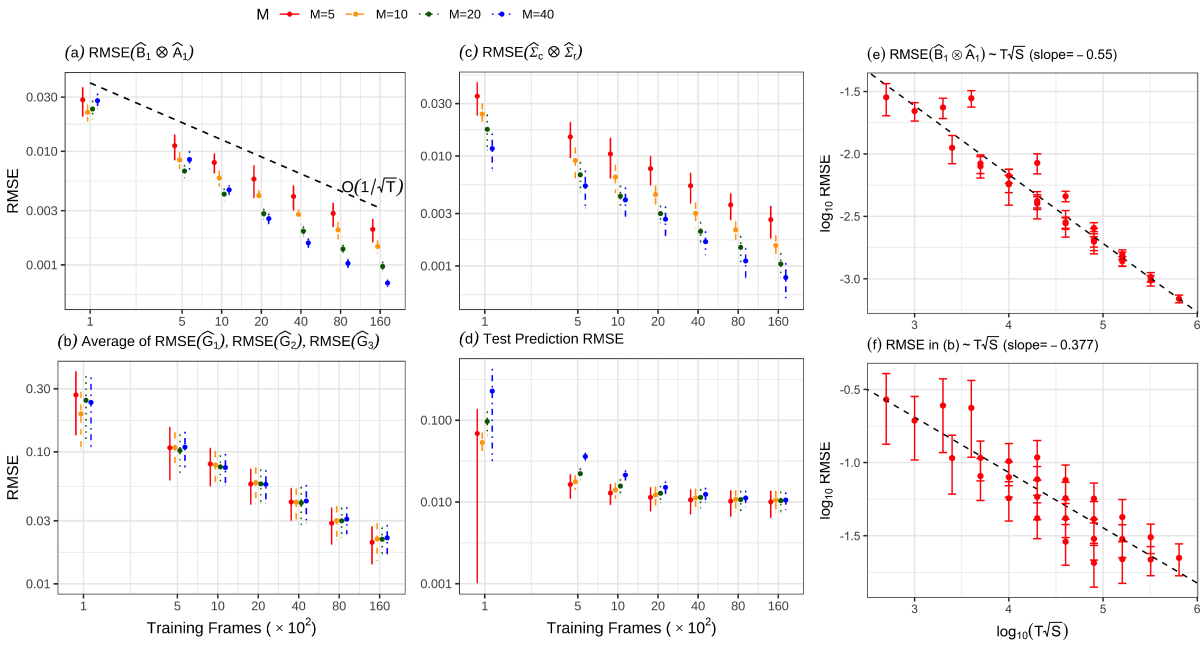


Figure 4.2: Panels (a), (b), (c) show the RMSE of the penalized MLE of the MARAC model. Panel (d) shows the testing set prediction RMSE subtracted by 1, where 1 is the noise variance of the simulated time series. Panels (a)-(d) have both axes plotted in $\log_{10}$ scale. (e) and (f) are RMSE of the autoregressive parameters and auxiliary covariates parameters under different $T\sqrt{S}$, plotted with both axes in $\log_{10}$ scale together with a fitted linear regression line.

The result shows that all model estimators are consistent and the convergence rate, under a fixed spatial dimensionality, is close to $1/\sqrt{T}$ (the black line in panel (a) shows a reference line of $O(1/\sqrt{T})$), echoing the result in Theorem 4.4.5. As the spatial dimensionality S increases, the RMSE for $\widehat{\mathbf{B}}_1 \otimes \widehat{\mathbf{A}}_1$ becomes even smaller, echoing the result in (4.24) and Table 4.1. The RMSE of the nonparametric estimators $\widehat{g}_1, \widehat{g}_2, \widehat{g}_3$, under a fixed spatial dimensionality, also decay at a rate of $1/\sqrt{T}$, echoing the result in Theorem 4.4.5 as well. The RMSE of the covariance matrix estimator $\widehat{\mathbf{\Sigma}}_c \otimes \widehat{\mathbf{\Sigma}}_r$ suggests that it is consistent, confirming the result of Proposition 4.4.4 and shows a convergence rate similar to $\widehat{\mathbf{B}}_1 \otimes \widehat{\mathbf{A}}_1$, though we did not provide the exact convergence rate theoretically.

In this simulation, we fix the variance of each element of $\mathrm{vec}(\mathbf{E}_t)$ to be unity. Therefore, the optimal testing set prediction RMSE should be unity. When plotting the test prediction RMSE in (d), we subtract 1 from all RMSE results and thus the RMSE should be interpreted as the RMSE for the *signal* part of the matrix time series. The test predic-

tion RMSE for all cases converges to zero, and for matrices of higher dimensionality, we typically require more training frames to reach the same prediction performance.

To validate the theoretical result of the high-dimensional MARAC in Theorem 4.4.8, we also plot the RMSE of $\widehat{\mathbf{B}}_1 \otimes \widehat{\mathbf{A}}_1$ and $\widehat{g}_1, \widehat{g}_2, \widehat{g}_3$ against $T\sqrt{S}$ in panels (e) and (f) of Figure 4.2. The trend line is fitted by linear regression, and it shows that $\widehat{\mathbf{B}}_1 \otimes \widehat{\mathbf{A}}_1$ converges roughly at the rate of $1/\sqrt{T}\sqrt[4]{S}$, which indicates that $c_{0,S} \asymp 1/\sqrt{S}$ under this specific simulation setup. It also shows that the functional parameter's convergence rate is around $(T\sqrt{S})^{-3/8}$, which coincides with our simulation setup where $r_0 \approx 3/4$ and the theoretical result in the last row of Table 4.1.

4.5.2 Lag Selection Consistency

In Section 4.3.2, we propose to select the lag parameters P and Q of the MARAC model using information criteria such as AIC and BIC. To validate the consistency of these model selection criteria, we simulate data from a MARAC$(2, 2)$ model with 5×5 matrix dimensionality. We consider a candidate model class with $1 \leq P, Q \leq 4$ and each model is fitted with $T \in \{1, 2, 4, 8\} \times 10^3$ frames with λ being chosen from a held-out validation set. In Table 4.2, we report the proportion of times that AIC and BIC select the correct P, Q individually (first two numbers in each parenthesis), and (P, Q) jointly (last number in each parenthesis) from 100 repetitions.

	$T = 1 \times 10^3$	$T = 2 \times 10^3$	$T = 4 \times 10^3$	$T = 8 \times 10^3$
AIC	$(.54, .99, .53)$	$(.55, .97, .53)$	$(.59, .96, .55)$	$(.65, .94, .59)$
BIC	$(1.00, .09, .09)$	$(.99, .56, .56)$	$(.97, .97, .94)$	$(.96, .99, .95)$

Table 4.2: Probability that AIC and BIC select the correct P (first number), Q (second number) and (P, Q) (third number) from 100 repetitions.

From Table 4.2, we find that AIC tends to select the model with more autoregressive lags but BIC performs consistently better under large sample sizes. This coincides with the findings in Hsu et al. (2021) for the matrix autoregression model.

4.5.3 Comparison with Alternative Methods

We compare our MARAC model against other competing methods for the matrix autoregression task. We simulate the matrix time series $\mathbf{X}_t$ from a MARAC(P, Q) model, with $P = Q \in \{1, 2, 3\}$, and the vector time series $\mathbf{z}_t \in \mathbb{R}^3$ from VAR(1). The dataset is generated with $T_{\text{train}} = T_{\text{val}} = T_{\text{test}} = 2000$. Under each (P, Q), we simulate with varying

matrix dimensionality with $M = N \in \{5, 10, 20, 40\}$. We evaluate the performance of each method via the testing set prediction RMSE. Each simulation scenario is repeated 20 times.

Under each P, Q, M, N specification, we consider the following five competing methods besides our own MARAC(P, Q) model.

1. MAR (Chen et al., 2021a):

$$\mathbf{X}_t = \sum_{p=1}^{P} \mathbf{A}_p \mathbf{X}_{t-p} \mathbf{B}_p^\top + \mathbf{E}_t, \mathbf{vec}(\mathbf{E}_t) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_c \otimes \mathbf{\Sigma}_r).$$

2. MAR with fixed-rank co-kriging (MAR+FRC) (Hsu et al., 2021):

$$\mathbf{X}_t = \sum_{p=1}^{P} \mathbf{A}_p \mathbf{X}_{t-p} \mathbf{B}_p^\top + \mathbf{E}_t, \mathbf{vec}(\mathbf{E}_t) \sim \mathcal{N}(\mathbf{0}, \sigma_\eta^2 \mathbf{I} + \mathbf{F}\mathbf{M}\mathbf{F}^\top),$$

   where $\mathbf{F} \in \mathbb{R}^{MN \times QD}$ is the multi-resolution spline basis (Tzeng and Huang, 2018).

3. MAR followed by a tensor-on-scalar linear model (MAR+LM) (Li and Zhang, 2017):

$$\mathbf{X}_t - \sum_{p=1}^{P} \widehat{\mathbf{A}}_p \mathbf{X}_{t-p} \widehat{\mathbf{B}}_p^\top = \sum_{q=1}^{Q} \boldsymbol{\mathcal{G}}_q \bar{\times} \mathbf{z}_{t-q} + \mathbf{E}_t, \mathbf{vec}(\mathbf{E}_t) \sim \mathcal{N}(\mathbf{0}, \sigma_\eta^2 \mathbf{I}), \tag{4.26}$$

   where $\widehat{\mathbf{A}}_p, \widehat{\mathbf{B}}_p$ come from a pre-trained MAR model and $\boldsymbol{\mathcal{G}}_q$ can be a low-rank tensor. The MAR+LM model can be considered as a two-step procedure for fitting the MARAC model.

4. Pixel-wise autoregression (Pixel-AR): for each $i \in [M], j \in [N]$, we have:

$$[\mathbf{X}_t]_{ij} = \alpha_{ij} + \sum_{p=1}^{P} \beta_{ijp} [\mathbf{X}_{t-p}]_{ij} + \sum_{q=1}^{Q} \boldsymbol{\gamma}_{ijq}^\top \mathbf{z}_{t-q} + [\mathbf{E}_t]_{ij}, \quad [\mathbf{E}_t]_{ij} \sim \mathcal{N}(0, \sigma_{ij}^2).$$

5. Vector Autoregression with Exogenous Predictor (VARX), which vectorizes the matrix time series and stacks them up with the vector time series as predictors.

The results of the average prediction RMSE obtained from the 20 repeated runs are plotted in Figure 4.3. Overall, our MARAC model outperforms the other competing methods under varying matrix dimensionality and lags. We make two additional remarks. First, when the matrix size is small (e.g., 5×5), the vector autoregression model (VARX) performs almost as well as the MARAC model and is better than other methods. However,

the performance of the VARX model gets worse quickly as the matrix becomes larger, indicating that sufficient dimension reduction is needed for dealing with large matrix time series. The MARAC model is a parsimonious version of VARX for such purposes. Secondly, the MAR, MAR with fixed-rank co-kriging (MAR+FRC), and two-step MARAC (MAR+LM) all perform worse than MARAC. This shows that when the auxiliary time series predictors are present, it is sub-optimal to remove them from the model (MAR), incorporate them implicitly in the covariance structure (MAR+FRC), or fit them separately in a tensor-on-scalar regression model (MAR+LM). Putting both matrix predictors and vector predictors in a unified framework like MARAC can be beneficial for improving prediction performances.

4.6 Application to Global Total Electron Content Forecast

For real data applications, we consider the problem of predicting the global total electron content (TEC) distribution, which we briefly introduce in Section 4.1. The TEC data we use is the IGS (International GNSS Service) TEC data, which are freely available from the National Aeronautics and Space Administration (NASA) Crustal Dynamics Data Information System (Hernández-Pajares et al., 2009). The spatial-temporal resolution of the data is $2.5°(\text{latitude}) \times 5°(\text{longitude}) \times 15(\text{minutes})$. We downloaded the data for September 2017, and the whole month of data form a matrix time series with $T = 2880$ and $M = 71, N = 73$. For the auxiliary covariates, we download the 15-minute resolution IMF Bz and Sym-H time series, which are parameters related to the near-Earth magnetic field and plasma (Papitashvili et al., 2014). We also download the daily F10.7 index, which measures the solar radio flux at 10.7 cm, as an additional auxiliary predictor. The IMF Bz and Sym-H time series are accessed from the OMNI dataset (Papitashvili and King, 2020) and the F10.7 index is accessed from the NOAA data repository (Tapping, 2013). These covariates measure the solar wind strengths. Strong solar wind might lead to geomagnetic storms that could increase the global TEC significantly.

We formulate our MARAC model for the TEC prediction problem as:

$$\text{TEC}_{t+h} = \sum_{p=1}^{P} \mathbf{A}_p \text{TEC}_{t-p} \mathbf{B}_p^\top + \sum_{q=1}^{Q} \boldsymbol{\mathcal{G}}_q \bar{\times} \mathbf{z}_{t-q} + \mathbf{E}_t, \tag{4.27}$$

where h is the forecast latency time and $\mathbf{z}_t \in \mathbb{R}^3$ includes the IMF Bz, Sym-H and F10.7 indices at time t. We consider the forecasting scenario with $h \in \{1, 2, \ldots, 24\}$, which corresponds to making forecasts from 15 minutes ahead up to 6 hours ahead. At each la-

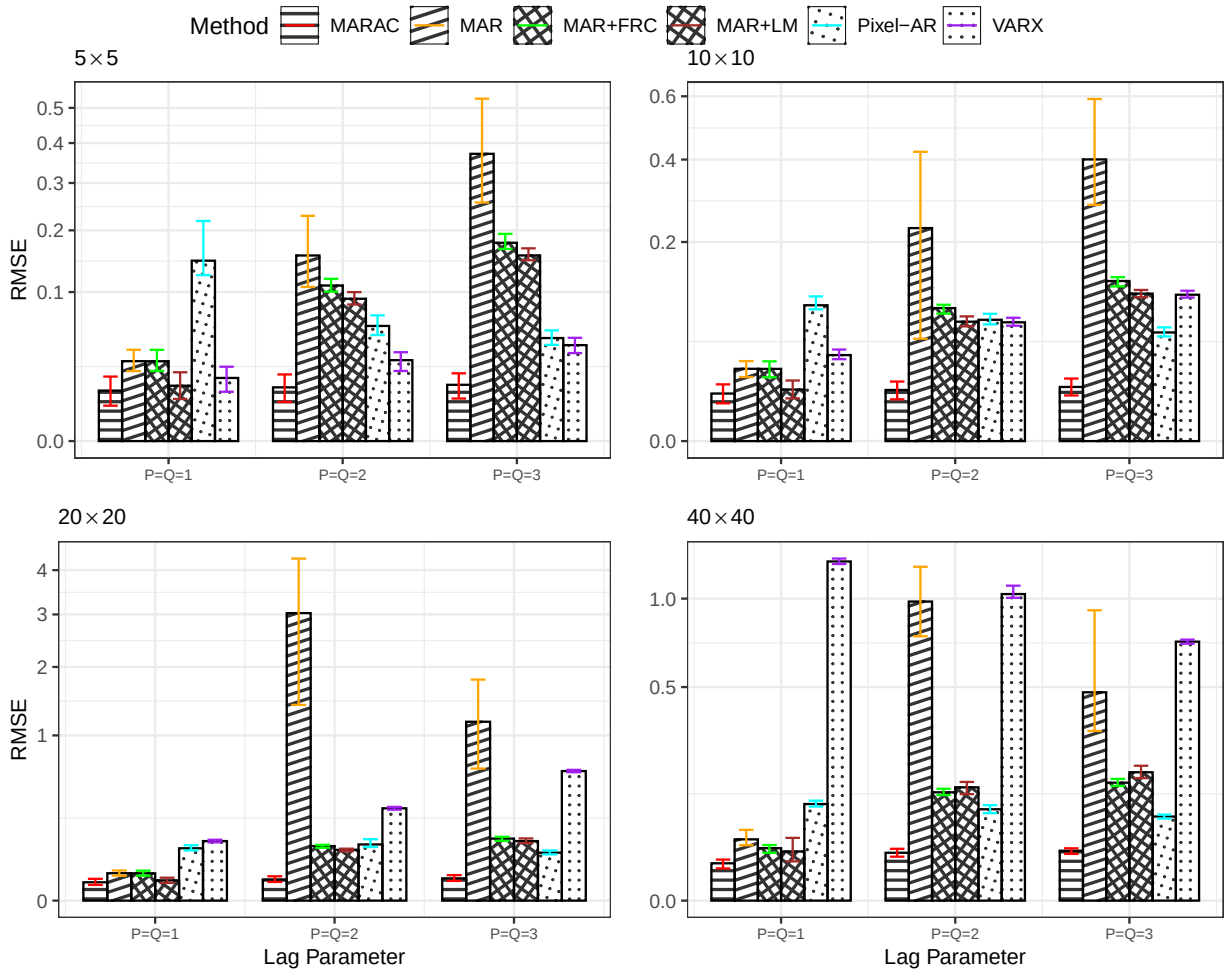


Figure 4.3: Testing set prediction RMSE comparison across six competing methods on the matrix autoregression task. Four panels correspond to four different matrix dimensionality (labeled on the top-left corner of each panel). Test prediction RMSE is subtracted by 1 for better visualization, where 1 is the noise variance of the simulated data. Error bar shows 95% CI of the 20 repeated runs. We rearrange the spacing between ticks along the y-axis using a square root transformation for better visualization.

tency time, we fit our MARAC(P, Q) model following (4.27) with $1 \leq P, Q \leq 3$. We fit the MARAC model with kernel truncation approximation using $R = 121$ basis functions from the truncated Lebedev kernel. As a comparison, we also fit the MAR model with $1 \leq P \leq 3$ and the MAR+LM model with $1 \leq P, Q \leq 3$, see the definition of MAR+LM model in (4.26). As a benchmark, we consider using TEC_{t-1} to predict TEC_{t+h} and name it the *persistence model*.

The $2,880$ frames of matrix data are split into a 70% training set, 15% validation set, and a 15% testing set following the chronological order. We choose the tuning parameter λ for MARAC based on the validation set prediction RMSE. The lag parameters P, Q are

chosen for all models based on the BIC. To increase computational speed, we assume that matrices $\mathbf{\Sigma}_r, \mathbf{\Sigma}_c$ are diagonal when fitting all models. We zero-meaned all sets of data using the mean of the matrix and vector time series of the training set.

In Figure 4.4(A), we report the pixel-wise prediction RMSE on the testing set. The result shows that when the latency time is low, the matrix autoregressive (MAR) model is sufficient for making the TEC prediction. As the latency time increases to around 4 to 5 hours, the auxiliary time series helps improve the prediction performance as compared to the MAR model. This coincides with the domain intuition that the disturbances from the solar wind to Earth's ionosphere will affect the global TEC distribution but with a delay in time of up to several hours. The additional prediction gain from incorporating the auxiliary covariates vanishes as one further increases the latency time, indicating that the correlation of the solar wind and global TEC is weak beyond a 6-hour separation.

In Figure 4.4(B), we visualize an example of the TEC prediction across the competing methods under the 4-hour latency time scenario (i.e., h=16). The MAR and MAR+LM results are similar and do not resemble the ground truth very well. The global TEC typically has two peaks located symmetrically around the equator, and both models fail to capture this as they provide a single patch in the middle. The MARAC model, however, can capture this fine-scale structure in its prediction. To further showcase the MARAC model prediction result, we decompose the prediction from the autoregressive component and the auxiliary covariates component and visualize them separately. The auxiliary covariate component highlights a sub-region in the southern hemisphere with high TEC values, complementing the prediction made by the autoregressive component.

4.7 Summary

In this chapter, we propose a new methodology for spatial-temporal matrix autoregression with non-spatial exogenous vector covariates. The model has an autoregressive component with bi-linear transformations on the lagged matrix predictors and an additive auxiliary covariate component with tensor-vector product between a tensor coefficient and the lagged vector covariates. We propose a penalized MLE estimation approach with a squared RKHS norm penalty and establish the estimator asymptotics under fixed and high matrix dimensionality. The model efficacy has been validated using both numerical experiments and an application to the global TEC forecast.

The application of our model can be extended to other spatial data with exogenous, non-spatial predictors and is not restricted to matrix-valued data but can be generalized to the tensor setting and potentially data without grid structure or containing missing data.

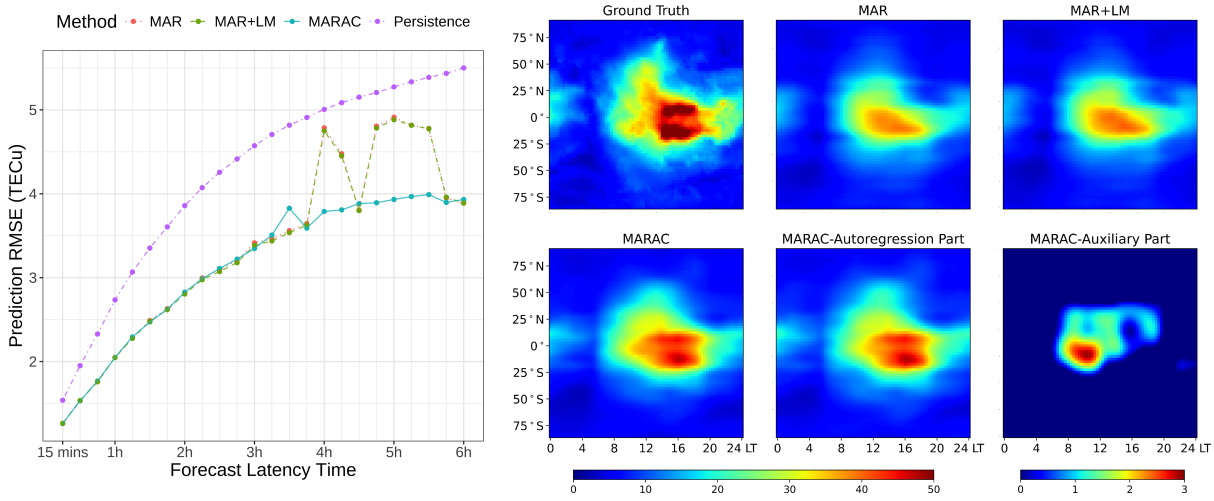


Figure 4.4: IGS TEC prediction results. Panel (A) shows the testing set prediction RMSE across four competing methods under 24 different latency times. Panel (B) shows an example of the predicted TEC at 10:45:00 UT, 2017-Sep-28, under the 4-hour latency time scenario. Note that the "MARAC-Auxiliary Part" plot has a different color bar underneath it and that color bar applies to it exclusively.

Furthermore, our model nests a simpler model that does not contain the autoregressive term, i.e. $P = 0$, and thus can be applied to matrix-on-scalar regression with spatial data.

There are several future directions related to our current work. Firstly, the current approach assumes that the tensor coefficient $\boldsymbol{\mathcal{G}}$ has spatial smoothness, and requires one to invert the kernel Gram matrix of size $S \times S$ which can lead to slow computation when the matrix dimensionality is high. The computational efficiency for $\boldsymbol{\mathcal{G}}$ determines the scalability of the current approach to large spatio-temporal datasets. Potential approaches for speeding up the computation include assuming that $\boldsymbol{\mathcal{G}}$ has a smooth and low-rank structure such as the smooth Tucker decomposition (Imaizumi and Hayashi, 2017) or using a better approximation than kernel truncation such as the nearest-neighbor processes (Datta et al., 2016) and the Vecchia approximation (Katzfuss and Guinness, 2021).

Secondly, the autoregressive coefficients $\mathbf{A}, \mathbf{B}$ are also high-dimensional when the matrix grows in size and additional assumptions on the sparsity of $\mathbf{A}, \mathbf{B}$ are necessary for scalable computation. There are several related works on this in both MAR and VAR model literature. Xiao et al. (2022) proposes a reduced-rank model that assumes $\mathbf{A}, \mathbf{B}$ are low-ranked, Guo et al. (2016) investigates banded autoregressive coefficients in the VAR model and this could be translated to banded $\mathbf{A}, \mathbf{B}$ in our model, Wang et al. (2022) imposes a low tensor rank structure over concatenated transition matrices of the VAR model and could be applied in our setting by assuming that $\boldsymbol{\mathcal{A}} = [\mathbf{A}_1, \ldots, \mathbf{A}_P]$, $\boldsymbol{\mathcal{B}} = [\mathbf{B}_1, \ldots, \mathbf{B}_P]$

are low-ranked tensors.

Finally, our work investigates a regression model with matrix response and a mixture of matrix and vector predictors, and we have found that the double asymptotics of the model estimators show a phase transition phenomenon and overall the estimating error is dominated by the matrix predictors. It is worthwhile to generalize the setup here to the regression model with tensor response and a mixture of predictors with arbitrary modes (e.g., vectors, matrices, and tensors) and investigate the interplay among predictors with non-uniform modes under a general theoretical framework.

CHAPTER 5

Scalar-on-Tensor Gaussian Process Regression with Contraction

5.1 Introduction

Regression models that deal with scalar labels and tensor covariates, i.e. scalar-on-tensor regression, have received widespread attention over the past decade (Hung and Wang, 2013; Zhou et al., 2013; Zhou and Li, 2014; Kang et al., 2018; Li et al., 2018; Papadogeorgou et al., 2021). Given m-mode tensor covariate $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_m}$ and scalar label $y \in \mathbb{R}$, the existing literature approaches the regression problem mainly via:

$$\mathbb{E}[y|\boldsymbol{\mathcal{X}}] = \alpha + \langle \boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{X}} \rangle \,, \tag{5.1}$$

where α is the intercept, $\boldsymbol{\mathcal{W}}$ is the regression coefficient tensor that matches the shape of $\boldsymbol{\mathcal{X}}$ and $\langle \cdot, \cdot \rangle$ denotes tensor inner product following Kolda and Bader (2009). This formulation can be readily adopted under the framework of generalized linear model (Zhou et al., 2013) while simultaneously preserving the tensor structure of $\boldsymbol{\mathcal{X}}$. Typically, tensor data is of ultra-high dimensions, and thus $\boldsymbol{\mathcal{W}}$ is also of high dimensionality. Various constraints have been introduced on $\boldsymbol{\mathcal{W}}$, such as tensor norm regularization (Guo et al., 2011; Zhou and Li, 2014) and tensor rank constraints (Papadogeorgou et al., 2021; Hao et al., 2021). These constraints induce a sparse and low-rank structure over $\boldsymbol{\mathcal{W}}$, making inferences of the high-order correlation between the scalar label and the tensor covariates tractable and interpretable.

Gaussian Process (GP) (Williams and Rasmussen, 2006) is an alternative approach to modeling complex correlation structures and has been applied to tensor regression problems (Kang et al., 2018), where a GP prior is imposed on $\boldsymbol{\mathcal{W}}$. In Yu et al. (2018), it is established that the tensor regression model (5.1), together with a low-rank constraint on $\boldsymbol{\mathcal{W}}$, leads to the same estimator $\widehat{\boldsymbol{\mathcal{W}}}$ as the tensor Gaussian Process (**Tensor-GP**) coupled

with a multi-linear kernel on the prior of $\boldsymbol{\mathcal{W}}$. A multi-linear kernel function $k(\cdot,\cdot)$ for m-mode tensors $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_m}$ can be defined in a Kronecker product form as:

$$k(\boldsymbol{\mathcal{X}}_i, \boldsymbol{\mathcal{X}}_j) = \mathbf{vec}(\boldsymbol{\mathcal{X}}_i)^\top \left(\otimes_{m'=1}^{m} \mathbf{K}_{m+1-m'}\right) \mathbf{vec}(\boldsymbol{\mathcal{X}}_j),$$

where $\mathbf{vec}(\cdot)$ is the vectorization operator and $\otimes$ denotes the matrix Kronecker product and $\mathbf{K}_1, \ldots, \mathbf{K}_m$ capture the mode-specific covariance structure of the regression coefficient tensor $\boldsymbol{\mathcal{W}}$ and are assumed to be low-rank. Interpreting this GP regression model can be hard since one needs to inspect the multi-linear kernel which deals with the tensor data at its original dimensionality $d = \prod_{m'=1}^{m} I_{m'}$.

The capability of the multi-linear Tensor-GP model to provide uncertainty quantification on the prediction makes it an attractive alternative to its counterpart in (5.1), but a sufficient dimension reduction on the tensor data is needed to make it more interpretable for scientific applications. In a different thread of literature, in Kossaifi et al. (2020), a tensor contraction operation is introduced before estimating the tensor regression model under the neural network settings. Instead of compressing the information of tensor data into a vector, the tensor data is contracted into a smaller *core* tensor with the same number of modes. Such a dimension reduction technique preserves the tensor structure of the data, making tensor regression or Tensor-GP directly applicable.

In this chapter, we propose a novel framework combining the merits of tensor contraction and Tensor-GP for the scalar-on-tensor regression task. Our framework consists of two major blocks. Firstly, we introduce tensor contraction to transform the tensor data $\boldsymbol{\mathcal{X}}$ to a feature tensor $\boldsymbol{\mathcal{Z}}$ with much lower dimensionality. Secondly, we apply the multi-linear Tensor-GP to the reduced-sized tensor $\boldsymbol{\mathcal{Z}}$ for regression. We build our model around a special type of tensor, i.e. the multi-channel imaging tensor, motivated by an application to astrophysical imaging analysis. Our model can be easily extended to a general tensor setup. We summarize our contributions as follows:

- We integrate tensor dimension reduction with Tensor-GP in a unified framework called **Tensor-GPST**, named after Tensor Gaussian Process with Spatial Transformation, allowing for learning a low-dimensional tensor representation in a supervised learning context.

- We propose to use the anisotropic total variation regularization (Wang et al., 2017) in the tensor contraction step for a sparse and spatially smooth tensor dimension reduction. We jointly estimate the parameters of Tensor-GPST under a penalized marginal likelihood approach coupled with the proximal gradient method (Parikh et al., 2014) with convergence guarantee.

5.2 Tensor Gaussian Process with Contraction

In this section, we will first introduce our method, namely Tensor-GPST, for the scalar-on-tensor regression task and then discuss the algorithm in Section 5.2.2 for estimating its parameters and conclude by discussing the theoretical guarantee of the algorithm convergence in Section 5.2.3. We follow the notations defined in Section 1.3 throughout this chapter.

5.2.1 Method

We consider a multi-channel imaging dataset $\{\boldsymbol{\mathcal{X}}_i, y_i\}_{i=1}^N$, where $\boldsymbol{\mathcal{X}}_i \in \mathbb{R}^{H\times W\times C}$ with H, W, C as the height, width, and number of channels, respectively; and $y_i \in \mathbb{R}$. We use $\mathbf{X}_i^{(c)} \in \mathbb{R}^{H\times W}, c \in [C]$ to denote the c^{th} channel of $\boldsymbol{\mathcal{X}}_i$. Gaussian process regression (GPR) (Williams and Rasmussen, 2006) specifies the prior for y_i as:

$$y_i = f(\boldsymbol{\mathcal{X}}_i) + \epsilon_i, \quad f(\cdot) \sim \text{GP}\left(m(\cdot), k(\cdot,\cdot)\right), \tag{5.2}$$

with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ being the idiosyncratic noise. The GP prior characterizes the unknown function $f(\cdot)$ evaluated at all data points as a multivariate Gaussian distribution, with a mean function $m(\cdot)$ and a covariance kernel function $k(\cdot,\cdot)$. Typically, $m(\cdot)$ is assumed to be zero and $k(\cdot,\cdot)$ fully specifies the behavior of the GP prior.

Given the high dimensionality of $\boldsymbol{\mathcal{X}}_i$, it would be difficult to directly estimate and interpret the tensor kernel $k(\cdot,\cdot)$. Here we consider adding one extra step called *tensor contraction*, which compresses the information of $\boldsymbol{\mathcal{X}}_i \in \mathbb{R}^{H\times W\times C}$ into a reduced-sized tensor $\boldsymbol{\mathcal{Z}}_i \in \mathbb{R}^{h\times w\times C}$, with $h < H, w < W$, via:

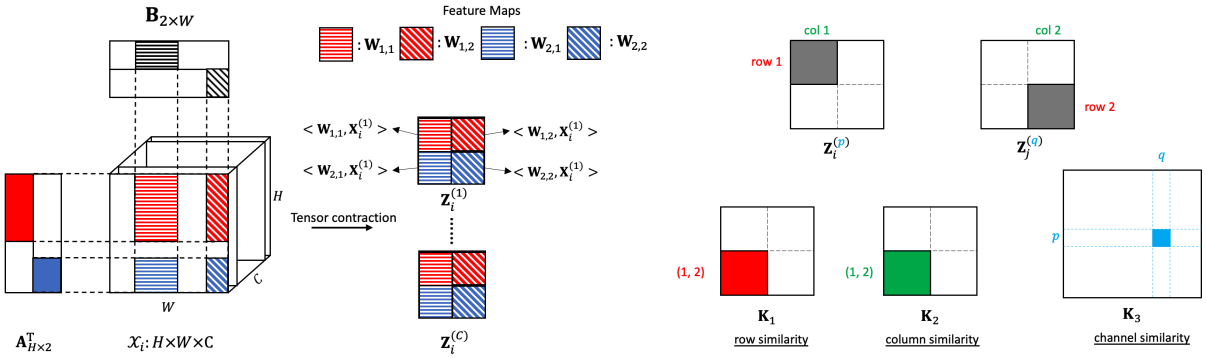
$$\boldsymbol{\mathcal{Z}}_i = g(\boldsymbol{\mathcal{X}}_i) = \boldsymbol{\mathcal{X}}_i \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{I}_C, \tag{5.3}$$

with $\mathbf{A} \in \mathbb{R}^{h\times H}, \mathbf{B} \in \mathbb{R}^{w\times W}$. In effect, $\mathbf{A}$ and $\mathbf{B}$ reduce the dimension of each channel of $\boldsymbol{\mathcal{X}}_i$ from $H \times W$ to $h \times w$ and one can rewrite (5.3) equivalently as:

$$\mathbf{Z}_i^{(c)} = \mathbf{A}\mathbf{X}_i^{(c)}\mathbf{B}^\top, \quad c = 1, 2, \ldots, C.$$

After applying (5.3) to $\boldsymbol{\mathcal{Z}}_i$, we then apply (5.2) on $\boldsymbol{\mathcal{Z}}_i$, as discussed later.

This formulation of tensor contraction can be found in a more general setting in tensor regression networks (Kossaifi et al., 2020), where tensor contraction can be applied to compress any tensors in a neural network. In our method, we envelope the tensor contraction operation within a tensor GP framework. Also, note that in (5.3), all channels share



(a) Tensor Contraction (b) Multi-Linear Kernel Interpretation

Figure 5.1: (a) Example of the tensor contraction step for tensor data $\boldsymbol{\mathcal{X}}_i \in \mathbb{R}^{H\times W\times C}$ to its latent tensor $\boldsymbol{\mathcal{Z}}_i \in \mathbb{R}^{2\times 2\times C}$. The tensor contracting factors $\mathbf{A}$, $\mathbf{B}$ are sparse (colored/dashed bands indicate nonzero elements) and they jointly extract features from $\mathbf{X}_i^{(1)}, \ldots, \mathbf{X}_i^{(C)}$ with rank-1 feature maps $\{\mathbf{W}_{1,1}, \mathbf{W}_{1,2}, \mathbf{W}_{2,1}, \mathbf{W}_{2,2}\}$. Each channel of $\mathbf{Z}_i^{(c)}$ has 2×2 features, based on the inner product of every feature map with the channel data $\mathbf{X}_i^{(c)}$. (b) Example of the multi-linear kernel with a pair of latent tensor data $(\boldsymbol{\mathcal{Z}}_i, \boldsymbol{\mathcal{Z}}_j)$. Any pair of pixels in $\boldsymbol{\mathcal{Z}}_i$ and $\boldsymbol{\mathcal{Z}}_j$, e.g., $\mathbf{Z}_i^{(p)}(1,1)$ and $\mathbf{Z}_j^{(q)}(2,2)$ in the plot (colored in gray), are weighted by the product of their row similarity $\mathbf{K}_1(1,1)$ (red), column similarity $\mathbf{K}_2(2,2)$ (green) and channel similarity $\mathbf{K}_3(p,q)$ (blue), in the kernel function (5.5) for defining the similarity of $\boldsymbol{\mathcal{Z}}_i, \boldsymbol{\mathcal{Z}}_j$. See (D.26) for a formulaic explanation.

the same tensor contracting factors $\mathbf{A}$ and $\mathbf{B}$, which preserves the spatial consistency of different channels of the reduced-sized tensor $\boldsymbol{\mathcal{Z}}$ for easier interpretation. Alternatively, one can replace the $\mathbf{I}_C$ in (5.3) with an arbitrary $C \times C$ matrix $\mathbf{C}$, ending up with the full tensor contraction in Kossaifi et al. (2020). We stick to (5.3) for simplicity in this chapter.

One can interpret the contracted tensor $\boldsymbol{\mathcal{Z}}_i$ as the latent low-dimensional representation of the original tensor $\boldsymbol{\mathcal{X}}_i$. Each $(s,t)^{\text{th}}$ element of $\mathbf{Z}_i^{(c)}$ is constructed via a matrix inner product with a rank-1 "feature map": $\mathbf{Z}_i^{(c)}(s,t) = \left\langle \boldsymbol{\alpha}_s^{\top}\boldsymbol{\beta}_t, \mathbf{X}_i^{(c)} \right\rangle$, where $\boldsymbol{\alpha}_s$ and $\boldsymbol{\beta}_t$, the basis of the feature map, are the s^{th} and t^{th} rows of $\mathbf{A}$ and $\mathbf{B}$, respectively. We denote the feature map $\left(\boldsymbol{\alpha}_s^{\top}\boldsymbol{\beta}_t\right)$ as $\mathbf{W}_{s,t} \in \mathbb{R}^{H\times W}$. A visual explanation of the tensor contraction operation is shown in Figure 5.1a. Note how elements of $\mathbf{Z}_i^{(c)}$ on the same row or column share the same feature map basis in $\mathbf{A}$ or $\mathbf{B}$.

Given the transformed tensor $\boldsymbol{\mathcal{Z}}_i = g(\boldsymbol{\mathcal{X}}_i)$, we assume a GP prior for $\mathbf{y} = (y_1, y_2, \ldots, y_N)^{\top}$ given $\boldsymbol{\mathcal{Z}}_1, \boldsymbol{\mathcal{Z}}_2, \ldots, \boldsymbol{\mathcal{Z}}_N$ with a multi-linear kernel (Yu et al., 2018):

$$y_i = h(\boldsymbol{\mathcal{Z}}_i) + \epsilon_i, \quad h(\cdot) \sim \text{GP}\left(0, k(\cdot,\cdot)\right), \tag{5.4}$$

where $k(\cdot,\cdot)$ is the multi-linear tensor kernel function:

$$k(\boldsymbol{\mathcal{Z}}_i, \boldsymbol{\mathcal{Z}}_j) = \mathbf{vec}(\boldsymbol{\mathcal{Z}}_i)^\top (\mathbf{K}_3 \otimes \mathbf{K}_2 \otimes \mathbf{K}_1)\, \mathbf{vec}(\boldsymbol{\mathcal{Z}}_j). \tag{5.5}$$

The multi-linear kernel defines a similarity metric between pairs of tensor data. We provide an illustration of the multi-linear kernel in Figure 5.1b. In this model, $\mathbf{K}_1 \in \mathbb{R}^{h\times h}, \mathbf{K}_2 \in \mathbb{R}^{w\times w}, \mathbf{K}_3 \in \mathbb{R}^{C\times C}$ capture the mode-specific covariance structure.

Combining (5.3), (5.4) and (5.5) together, our method essentially specifies the following tensor Gaussian Process with a new kernel $\mathcal{K}(\cdot,\cdot)$:

$$y_i = f(\boldsymbol{\mathcal{X}}_i) + \epsilon_i, \quad f(\cdot) \sim \mathrm{GP}\left(0, \mathcal{K}(\cdot,\cdot)\right), \tag{5.6}$$

$$\mathcal{K}\left(\boldsymbol{\mathcal{X}}_i, \boldsymbol{\mathcal{X}}_j\right) = \mathbf{vec}(\boldsymbol{\mathcal{X}}_i)^\top \left(\mathbf{K}_3 \otimes \mathbf{K}_2^* \otimes \mathbf{K}_1^*\right) \mathbf{vec}(\boldsymbol{\mathcal{X}}_j), \tag{5.7}$$

$$\mathbf{K}_2^* = \mathbf{B}^\top \mathbf{K}_2 \mathbf{B}, \quad \mathbf{K}_1^* = \mathbf{A}^\top \mathbf{K}_1 \mathbf{A}, \tag{5.8}$$

and we call the framework **Tensor** **G**aussian **P**rocess with **S**patial **T**ransformation (**Tensor-GPST**), where $\mathbf{A}$ and $\mathbf{B}$ transform, in a bi-linear way, the spatial information contained in the imaging data.

Another way of expressing the model is via tensor regression (5.1) on the original tensor $\boldsymbol{\mathcal{X}}$. Equivalently, we assume a Gaussian prior over $\boldsymbol{\mathcal{W}}$:

$$\begin{aligned} &\mathbf{vec}(\boldsymbol{\mathcal{W}}) \sim (\mathbf{I}_C \otimes \mathbf{B} \otimes \mathbf{A})^\top \mathbf{vec}(\boldsymbol{\mathcal{T}}), \boldsymbol{\mathcal{T}} \in \mathbb{R}^{h\times w\times C}, \\ &\mathbf{vec}(\boldsymbol{\mathcal{T}}) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{K}_3 \otimes \mathbf{K}_2 \otimes \mathbf{K}_1\right), \end{aligned} \tag{5.9}$$

which is similar to a tensor factor model (Chen et al., 2024) coupled with a Gaussian factor with Kronecker-product covariance structure.

5.2.2 Estimating Algorithm

To estimate the model parameters of Tensor-GPST in (5.6), (5.7), (5.8), including the tensor contracting factors $\mathbf{A}, \mathbf{B}$, the multi-linear kernel factors $\mathbf{K}_1, \mathbf{K}_2, \mathbf{K}_3$, and the idiosyncratic noise variance σ^2, we minimize the negative marginal Gaussian log-likelihood $\ell(\mathbf{y}|\mathbf{A}, \mathbf{B}, \mathbf{K}_1, \mathbf{K}_2, \mathbf{K}_3, \sigma)$:

$$\ell(\mathbf{y}|\mathbf{A}, \mathbf{B}, \mathbf{K}_1, \mathbf{K}_2, \mathbf{K}_3, \sigma) = \frac{1}{2}\log|\mathbf{K} + \mathbf{D}_\sigma| + \frac{1}{2}\mathbf{y}^\top (\mathbf{K} + \mathbf{D}_\sigma)^{-1}\mathbf{y} + \text{constant}, \tag{5.10}$$

where $\mathbf{K}$ is an $N \times N$ empirical kernel gram matrix computed using the kernel function (5.7) for all pairs of tensor data and $\mathbf{D}_\sigma = \sigma^2 \mathbf{I}_N$.

To speed up the computation, we approximate each multi-linear kernel factor with a factorized form:

$$\mathbf{K}_1 = \mathbf{U}_1^\top \mathbf{U}_1, \mathbf{K}_2 = \mathbf{U}_2^\top \mathbf{U}_2, \mathbf{K}_3 = \mathbf{U}_3^\top \mathbf{U}_3, \tag{5.11}$$

where $\mathbf{U}_1 \in \mathbb{R}^{r_1 \times h}, \mathbf{U}_2 \in \mathbb{R}^{r_2 \times w}, \mathbf{U}_3 \in \mathbb{R}^{r_3 \times C}$. $\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3$ are orthogonal matrices with $r_1 \leq h, r_2 \leq w, r_3 \leq C$. The tuning parameter is set as such that $r_1 = h, r_2 = w, r_3 = C$ throughout the chapter but can be set to smaller values to enforce a low-rank constraint. With the factorization assumption, one can decompose the gram matrix $\mathbf{K}$ as $\widetilde{\mathbf{U}}\widetilde{\mathbf{U}}^\top$, where:

$$\widetilde{\mathbf{U}} = \widetilde{\boldsymbol{\mathcal{X}}}^\top \left(\mathbf{I}_C \otimes \mathbf{B} \otimes \mathbf{A}\right)^\top \left(\mathbf{U}_3 \otimes \mathbf{U}_2 \otimes \mathbf{U}_1\right)^\top,$$

and $\widetilde{\boldsymbol{\mathcal{X}}} = [\mathbf{vec}(\boldsymbol{\mathcal{X}}_1); \mathbf{vec}(\boldsymbol{\mathcal{X}}_2); \ldots; \mathbf{vec}(\boldsymbol{\mathcal{X}}_N)]$. The factorized form of $\mathbf{K}$ can simplify the computation of the gradients since one can invert the covariance matrix $(\mathbf{K} + \mathbf{D}_\sigma)$ with the Woodbury identity, as shown in Appendix D.2. The computational complexity of the algorithm is thus reduced from the canonical $\mathcal{O}(N^3)$ to $\mathcal{O}(N^2 D)$, where $D = HWC$ is the dimension of the data tensor.

Since the tensor contracting factors $\mathbf{A}, \mathbf{B}$ are extracting spatial features from each channel of $\boldsymbol{\mathcal{X}}_i$, we assume that each spatial feature can be constructed from several spatially-contiguous regions for better interpretability. This leads us to the assumption that each feature map $\mathbf{W}_{s,t} = \boldsymbol{\alpha}_s^\top \boldsymbol{\beta}_t$ has certain degrees of spatial smoothness. We introduce the spatial smoothness assumption into our model via regularizing its anisotropic total variation norm $\|\mathbf{W}_{s,t}\|_{\text{TV}}$, which is defined as:

$$\begin{aligned}\|\mathbf{W}_{s,t}\|_{\text{TV}} = &\sum_{i=1}^{H-1} \sum_{j=1}^{W} |\mathbf{W}_{s,t}(i+1, j) - \mathbf{W}_{s,t}(i, j)| \\ &+ \sum_{i=1}^{H} \sum_{j=1}^{W-1} |\mathbf{W}_{s,t}(i, j+1) - \mathbf{W}_{s,t}(i, j)|\,.\end{aligned}$$

A more general class of total variation norm penalty on tensor regression model coefficients can be found in Wang et al. (2017). In Lemma 5.2.1, we derive a simplified form of $\|\mathbf{W}_{s,t}\|_{\text{TV}}$, making the estimation of $\mathbf{A}$ and $\mathbf{B}$ easier in later steps.

Lemma 5.2.1. *The anisotropic total variation (TV) norm on feature map $\{\mathbf{W}_{s,t}\}_{s=1,t=1}^{h,w}$ induces a fused-lasso (Tibshirani et al., 2005) penalty on $\mathbf{A}$ (and $\mathbf{B}$), namely:*

$$\sum_{s=1}^{h} \sum_{t=1}^{w} \|\mathbf{W}_{s,t}\|_{\text{TV}} = \|\nabla_x \mathbf{B}\|_1 \|\mathbf{A}\|_1 + \|\mathbf{B}\|_1 \|\nabla_x \mathbf{A}\|_1\,, \tag{5.12}$$

where ∇_x *computes the horizontal gradient of a matrix, i.e.* $\nabla_x \mathbf{A}_{m\times n}(i,j) = \mathbb{1}_{\{j\neq n\}}[\mathbf{A}(i,j+1) - \mathbf{A}(i,j)]$*, and* $\|\cdot\|_1$ *is the elementwise* ℓ_1*-norm of a matrix.*

We leave the proof to Appendix D.1.

The fused-lasso penalty penalizes the sparsity and smoothness of $\mathbf{A}$, weighted by the smoothness and sparsity of $\mathbf{B}$, and vice versa. Jointly, our estimating problem is attempting to minimize the following penalized negative log-likelihood:

$$L(\mathbf{y}|\mathbf{A},\mathbf{B},\mathbf{U}_{1:3},\sigma) = \ell(\mathbf{y}|\mathbf{A},\mathbf{B},\mathbf{U}_{1:3},\sigma) + \lambda \mathrm{R}(\mathbf{A},\mathbf{B}), \tag{5.13}$$

where $\mathrm{R}(\mathbf{A},\mathbf{B}) = \|\nabla_x \mathbf{B}\|_1 \|\mathbf{A}\|_1 + \|\mathbf{B}\|_1 \|\nabla_x \mathbf{A}\|_1$ and $\mathbf{U}_{1:3}$ is the collection of $\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3$.

The total variation penalty can create feature maps with sharp edges and leads to sparsity for interpretation. In the estimating algorithm, we use proximal gradient descent to estimate the tensor contracting factors $\mathbf{A}, \mathbf{B}$ and cyclically update the parameters in the order of: $\mathbf{A} \to \mathbf{B} \to (\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3) \to \sigma \to \mathbf{A} \to \ldots$. The fused-lasso penalty over $\mathbf{A}$ and $\mathbf{B}$ makes the proximal step a well-defined *fused lasso 1-D signal approximation* problem (Friedman et al., 2007). Specifically, at the $(i+1)^{\text{th}}$ iteration, we first propose a gradient descent update for $\mathbf{A}$, denoted as $\widehat{\mathbf{A}}^{(i+\frac{1}{2})}$, with step size η_i. The final updated value for $\mathbf{A}$, i.e. $\widehat{\mathbf{A}}^{(i+1)}$, is the minimizer of the proximal step:

$$\begin{aligned}\widehat{\mathbf{A}}^{(i+1)} &= \mathbf{prox}_{\mathrm{TV}}\left(\widehat{\mathbf{A}}^{(i+\frac{1}{2})}\right) \\ &= \arg\min_{\mathbf{A}} \left\{ \frac{1}{2\eta_i} \left\| \mathbf{A} - \widehat{\mathbf{A}}^{(i+\frac{1}{2})} \right\|_{\mathrm{F}}^2 + \lambda \mathrm{R}(\mathbf{A}, \widehat{\mathbf{B}}^{(i)}) \right\},\end{aligned}$$

which can be easily solved by first solving the minimization without the ℓ_1-penalty on $\mathbf{A}$ and then apply a soft-thresholding operator to obtain the exact minimizer (see Proposition 1 of Friedman et al. (2007) for the justification). The same procedure applies when one updates $\mathbf{B}$. We summarize the outline of the estimating algorithm in Algorithm 5.1 and provide the details of the derivation of gradients and the proximal step in Appendix D.2.

Since any pair of $(\mathbf{A},\mathbf{B})$ can be re-scaled by a constant c_1 such that: $(\mathbf{B}\otimes\mathbf{A}) = (c_1^{-1}\mathbf{B})\otimes(c_1\mathbf{A})$, we re-scale the norm of $(\widehat{\mathbf{A}}^{(i)}, \widehat{\mathbf{B}}^{(i)})$ after each iteration to ensure that there is no scaling identifiability issue for the tensor contraction operation.

We do not enforce the orthonormality of $\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3$, but a good initialization can still obtain reasonable approximations according to Yu et al. (2018). To give a warm start of the model parameters, one can consider solving a tensor regression problem and a tucker

decomposition problem subsequently, as inspired by (5.9):

$$\min_{\substack{\boldsymbol{\mathcal{T}}\in\mathbb{R}^{h\times w\times C}\\ \mathbf{A}\in\mathbb{R}^{h\times H}\\ \mathbf{B}\in\mathbb{R}^{w\times W}}} \sum_{i=1}^{N}\left(y_i-\left\langle \boldsymbol{\mathcal{X}}_i, \boldsymbol{\mathcal{T}}\times_1\mathbf{A}^\top\times_2\mathbf{B}^\top\right\rangle\right)^2, \tag{5.14}$$

$$\min_{\substack{\boldsymbol{\mathcal{S}}\in\mathbb{R}^{r_1\times r_2\times r_3}\\ \mathbf{U}_1,\mathbf{U}_2,\mathbf{U}_3}} \left\|\widehat{\boldsymbol{\mathcal{T}}}-\boldsymbol{\mathcal{S}}\times_1\mathbf{U}_1^\top\times_2\mathbf{U}_2^\top\times_3\mathbf{U}_3^\top\right\|^2. \tag{5.15}$$

One obtains $\widehat{\mathbf{A}}^{(0)},\widehat{\mathbf{B}}^{(0)}$ from (5.14) and $\widehat{\mathbf{U}}_{1:3}^{(0)}$ from (5.15).

Algorithm 5.1 Alternating Proximal Gradient Descent Algorithm for Tensor-GPST

Input: Random initialization: $\widehat{\mathbf{A}}^{(0)},\widehat{\mathbf{B}}^{(0)},\widehat{\mathbf{U}}_1^{(0)},\widehat{\mathbf{U}}_2^{(0)},\widehat{\mathbf{U}}_3^{(0)},\widehat{\sigma}^{(0)}$.

Set iteration counter $i\leftarrow 0$.
while not converge and $i\le$ max-iter **do**
  $\widehat{\mathbf{A}}^{(i+\frac{1}{2})}\leftarrow\widehat{\mathbf{A}}^{(i)}-\eta_i\nabla_{\mathbf{A}}\ell(\mathbf{y}|\widehat{\mathbf{A}}^{(i)},\widehat{\mathbf{B}}^{(i)},\widehat{\mathbf{U}}_{1:3}^{(i)},\widehat{\sigma}^{(i)})$.
  $\widehat{\mathbf{A}}^{(i+1)}\leftarrow\mathbf{prox}_{\mathrm{TV}}(\widehat{\mathbf{A}}^{(i+\frac{1}{2})})$. % Fused-Lasso
  $\widehat{\mathbf{B}}^{(i+\frac{1}{2})}\leftarrow\widehat{\mathbf{B}}^{(i)}-\eta_i\nabla_{\mathbf{B}}\ell(\mathbf{y}|\widehat{\mathbf{A}}^{(i+1)},\widehat{\mathbf{B}}^{(i)},\widehat{\mathbf{U}}_{1:3}^{(i)},\widehat{\sigma}^{(i)})$.
  $\widehat{\mathbf{B}}^{(i+1)}\leftarrow\mathbf{prox}_{\mathrm{TV}}(\widehat{\mathbf{B}}^{(i+\frac{1}{2})})$. % Fused-Lasso
  Re-scale $\widehat{\mathbf{A}}^{(i+1)},\widehat{\mathbf{B}}^{(i+1)}$ s.t. $\|\widehat{\mathbf{A}}^{(i+1)}\|_{\mathrm{F}}=1$.
  for j=1:3 **do**
    $\mathbf{G}_j\leftarrow\nabla_{\mathbf{U}_j}\ell(\mathbf{y}|\widehat{\mathbf{A}}^{(i+1)},\widehat{\mathbf{B}}^{(i+1)},\widehat{\mathbf{U}}_{-j}^{(i)},\widehat{\sigma}^{(i)})$. % $\mathbf{U}_{-j}$ includes $\mathbf{U}_1,\mathbf{U}_2,\mathbf{U}_3$ but $\mathbf{U}_j$.
    $\widehat{\mathbf{U}}_j^{(i+1)}\leftarrow\widehat{\mathbf{U}}_j^{(i)}-\eta_i\mathbf{G}_j$.
  end for
  $t\leftarrow\nabla_\sigma\ell(\mathbf{y}|\widehat{\mathbf{A}}^{(i+1)},\widehat{\mathbf{B}}^{(i+1)},\widehat{\mathbf{U}}_{1:3}^{(i+1)},\widehat{\sigma}^{(i)})$.
  $\widehat{\sigma}^{(i+1)}\leftarrow\widehat{\sigma}^{(i)}-\eta_i t$.
  $i\leftarrow i+1$.
end while

Output: Estimators for Tensor-GPST: $\widehat{\mathbf{A}}^{(i)},\widehat{\mathbf{B}}^{(i)},\widehat{\mathbf{U}}_{1:3}^{(i)},\widehat{\sigma}^{(i)}$.

We typically implement Algorithm 5.1 with a constant step size and terminate the algorithm after a pre-specified number of iterations. If the algorithm does not converge, we will try multiple different random initialization and set the resulting model to the one with the minimal loss.

5.2.3 Convergence Analysis

In this subsection, we provide the convergence analysis of Algorithm 5.1. Theorem 5.2.2 provides the upper bound of the loss function (5.13), evaluated at the estimators output by the algorithm, with respect to its global minimum. We show that the total variation

penalty and the alternating proximal gradient descent introduce extra gaps between the achieved loss and its global minimum.

Theorem 5.2.2. *Given the loss function $L(\cdot)$ in (5.13), assume that the negative log-likelihood $\ell(\cdot)$ is convex for any of the four parameter blocks: $\{\mathbf{A}\}, \{\mathbf{B}\}, \{\mathbf{U}_{1:3}\}, \{\sigma\}$, with the other three blocks being fixed, and the gradients of $\ell(\cdot)$ are Lipschitz continuous with Lipschitz constant: $M_{\mathbf{A}}, M_{\mathbf{B}}, M_{\mathbf{U}}, M_\sigma$, respectively. Then with a constant stepsize $\alpha \leq 1/\max\{M_{\mathbf{A}}, M_{\mathbf{B}}, M_{\mathbf{U}}, M_\sigma\}$, the alternating proximal gradient descent algorithm in Algorithm 5.1 leads to the following upper bound on the loss function $L(\cdot)$:*

$$4\,(K+1)\left[L(\widehat{\boldsymbol{\theta}}^{(K+1)}) - L(\boldsymbol{\theta}^*)\right] \leq \frac{\delta^{(0)}}{2\alpha} + \sum_{k=0}^{K} h_\lambda(\widehat{\mathbf{A}}^{(k+1)} - \mathbf{A}^*, \widehat{\mathbf{B}}^{(k+1)} - \mathbf{B}^*, \widehat{\mathbf{B}}^{(k)} - \mathbf{B}^*) + \frac{1}{2\alpha}\sum_{k=0}^{K} \tau\left(\widehat{\boldsymbol{\theta}}^{(k+1)}, \widehat{\boldsymbol{\theta}}^{(k)}, \boldsymbol{\theta}^*\right), \tag{5.16}$$

where $\widehat{\boldsymbol{\theta}}^{(k)} = \left\{\widehat{\mathbf{A}}^{(k)}, \widehat{\mathbf{B}}^{(k)}, \widehat{\mathbf{U}}_{1:3}^{(k)}, \widehat{\sigma}^{(k)}\right\}$ and $\boldsymbol{\theta}^*$ is the global minimizer of $L(\cdot)$. $\delta^{(0)}$ is the squared ℓ_2 initialization error, $h_\lambda(\cdot) \geq 0$ is the total-variation gap (TV-gap) and $\tau(\cdot) \geq 0$ is the alternating descent gap (ALT-gap). K is the total number of iterations.

As a result of (5.16), if one has any three blocks of parameters fixed at their global minima, the remaining block will converge to its global minima at the rate of $\mathcal{O}(1/K)$, which echoes the convergence rate of (proximal) gradient descent. We leave the proof to Appendix D.3 and make a few remarks.

Remark 5.2.3. *As $\widehat{\mathbf{A}}^{(k)} \to \mathbf{A}^*, \widehat{\mathbf{B}}^{(k)} \to \mathbf{B}^*$, one has $h_\lambda(\cdot) \to 0$. The TV-gap is incurred because we alternatively update $\mathbf{A}$ and $\mathbf{B}$, and using the current iteration's estimate of $\mathbf{A}$ (or $\mathbf{B}$) for updating $\mathbf{B}$ (or $\mathbf{A}$) with the total variation penalty leads to extra errors. See the definition of $h_\lambda(\cdot)$ in (D.23).*

Remark 5.2.4. *As $\widehat{\boldsymbol{\theta}}^{(k)} \to \boldsymbol{\theta}^*$, $\tau(\cdot) \to 0$. The ALT-gap $\tau(\cdot)$ arises because we use the current iteration's estimate for all but one block of parameters to estimate the gradient of the block of interest. If the algorithm terminates at a local minima, the non-vanishing TV-gap and ALT-gap leads to a non-zero gap for the achieved loss from the global minimum. See the definition of $\tau(\cdot)$ in (D.24).*

Remark 5.2.5. *Tensor regression models with Tucker-type low-rankness have non-convex negative-likelihood function $\ell(\cdot)$ (Li et al., 2018). But conditioning on all but one block of parameter, $\ell(\cdot)$ is convex for each individual block. We do not verify the convexity of $\ell(\cdot)$ in our particular model due to the complexity of the kernel function. Empirically, as we show in Figure D.1 in Appendix D.3 and also demonstrated in Yu et al. (2018), such alternating gradient descent algorithm works well with the optimization problem and the loss function decays at the rate of $\mathcal{O}(1/K)$.*

5.3 Experiments

In this section, we validate our method via both simulation studies and an application to the solar flare dataset. We also compare our method against other benchmark tensor regression models. In particular, we are interested in applications to imaging data where the predictive signals appear in different channels and various locations within a channel. Such patterns are common in astrophysical imaging data where the solar flare precursors could appear in the images collected by the astrophysical telescopes at various frequencies and the locations of the precursors could be random within an image channel.

5.3.1 Simulation Study

We simulate a tensor dataset $\{\boldsymbol{\mathcal{X}}_i\}_{i=1}^N$ with each $\boldsymbol{\mathcal{X}}_i$ having 3 channels of size 25×25. For each $25 \times 25 \times 3$ tensor data $\boldsymbol{\mathcal{X}}_i$, we randomly pick one of the three channels as the *signal* channel, with equal probability, and the remaining two channels as the *noise* channels. The noise channel contains i.i.d. pixels from $\mathcal{N}(0, 0.3)$, and the signal channel uses the same background noise distribution except having a 5×5 *signal* block that contains i.i.d. pixels from $\mathcal{N}(4, 0.3)$. The location of the 5×5 block is fixed at the center of the 25×25 image if channel 2 is the signal channel (see Type 2 in Figure 5.2), and is randomly picked at one of the four corners (top-left, top-right, bottom-left, bottom-right) if channel 1 or 3 is the signal channel (see Type 1 and 3 in Figure 5.2).

We simulate the tensor contracting factors $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{3\times 25}$ with a banded structure, leading to a $3 \times 3 \times 3$ contracted tensor $\boldsymbol{\mathcal{Z}}$, such that $\mathbf{A}$ and $\mathbf{B}$ are extracting features from the 5×5 blocks with *signal*, see the bottom of Figure 5.2 for the example of the contracted tensor $\boldsymbol{\mathcal{Z}}$. The multi-linear kernel setup and the generating process of the regression labels $\{y_i\}_{i=1}^N$ are detailed in Appendix D.4. Generally, channel 2 is simulated such that it is negatively correlated with channels 1 & 3, and channels 1 & 3 are nearly perfectly correlated. As a result, Type 1 & 3 tensors have similar regression labels and differ from those of Type 2.

With the simulation setups, we compare our model against these baseline tensor regression models: Tensor-GP (**GP**) (Yu et al., 2018), ℓ_2-regularized tensor regression with CANDECOMP/PARAFAC (CP) tensor rank constraints (Guo et al., 2011) (**CP**), ℓ_2-regularized tensor regression with coefficient tensor following a Tucker decomposition (Li et al., 2018) (**Tucker**). In order to check the sensitivity of the choice of the kernel, we also fit the GP model to the vectorized tensor data with a squared-exponential kernel (**SE**). To showcase the effectiveness of tensor contraction, we also consider fitting a model with a tensor contraction step followed by a GP with squared-exponential kernel for the vector-

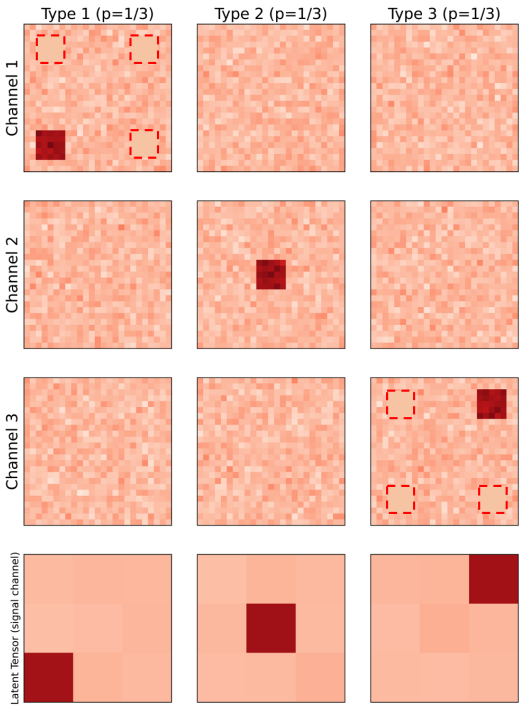


Figure 5.2: Three types of the simulated tensor data ($\boldsymbol{\mathcal{X}}_i \in \mathbb{R}^{25\times25\times3}$). Each column is a type (Type 1,2,3) and every sample has equal probability of being one of the three types. Each row (row 1-3) is a data channel (channel 1,2,3). Type 1, 2 and 3 have their *signal* channel in channel 1, 2 and 3, respectively. But the location of the 5×5 signal block is positioned differently. Type 2 has the signal fixed at the center, while type 1 and 3 has the signal placed, with equal probability, in one of the four corners (dashed block shows the other three possible locations). Samples shown are one realization of the simulation. The latent tensor $\boldsymbol{\mathcal{Z}}$'s signal channel is shown at the bottom. See details in Appendix D.4.

ized, reduced-sized tensor (**SE+TC**). For simplicity, we implement **SE+TC** by training the model, without the total variation penalty, in an end-to-end fashion with the `GPyTorch` and `Tensorly-Torch` packages in `Python`. Both models involving the SE kernel have automatic relevance determination (ARD) length scales (Bishop and Nasrabadi, 2006).

We simulate the data with size $N \in \{200, 500\}$ and use 75% for training and 25% for testing and compare the rooted-mean-squared-error (RMSE) on both training and testing across all models above as well as our own Tensor-GPST model (**GPST**). We set the latent tensor dimension as $3 \times 3 \times 3$ for **GPST** and **SE+TC** and the rank for $\mathbf{K}_1, \mathbf{K}_2, \mathbf{K}_3$ of **GP** as 3 and the CP rank as 9 for **CP** and the multi-linear rank as $3 \times 3 \times 3$ for **Tucker** such that the low-rankness is comparable across all methods. We select the regularization tuning parameter for all models with hyperparameters by 5-fold cross validation. The simulation experiment is iterated 10 times and the testing RMSE is shown in Table 5.1.

Model	$N = 200$	$N = 500$
GP	0.728±0.125	0.664±0.131
CP	0.550±0.100	0.548±0.054
Tucker	0.589±0.206	0.568±0.107
SE	2.504±2.672	3.275±4.204
SE+TC	0.627±0.169	0.587±0.098
GPST (Our Method)	0.578±0.107	0.552±0.076

Table 5.1: Test prediction RMSE for simulated data for various tensor regression models. 95% confidence interval after $\pm$. Results are based on 10 repeated runs.

The Tensor-GP (**GP**) method has relatively worse performance on the testing set compared to other low-rank tensor regression methods such as **CP** and **Tucker**. Our method, namely **GPST**, achieves similar performance to the low-rank tensor regression methods (not statistically significantly worse). The GP with vectorized tensor data and squared-exponential kernel, namely **SE**, performs extremely poorly, which reveals the fact that by vectorizing tensor data one loses the essential structural information of the data. This result necessitates the choice of kernel that is suitable for tensor data, such as the tensor GP. After adding an extra tensor contraction step, the GP with squared-exponential kernel (i.e. **SE+TC**) performs relatively close to the low-rank tensor regression methods as well as our **GPST** and is better than the tensor GP. This further suggests that regardless of the kernel choice, the tensor contraction step can boost the performance of GP regression models with tensor covariates. Effectively, the tensor contraction step extracts useful features from the original tensor data for regression, so even if one vectorizes the reduced-sized tensor, one does not lose as much information as the case where tensor contraction is not being used. Finally, we note that with a large sample size ($N = 500$), the prediction RMSE of the test set gets smaller for all methods but **SE**.

To make further comparisons of the variants of different Gaussian Process models listed in Table 5.1 on their ability to quantify the uncertainties of the predictions made, we compare these GP models' mean standardized log loss (MSLL) (Williams and Rasmussen, 2006), as defined below:

$$\text{MSLL} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \left\{ \frac{1}{2} \log\left(2\pi\widehat{\sigma}^2\right) + \frac{(y_i - \widehat{y}_i)^2}{2\widehat{\sigma}^2} \right\},$$

where $\widehat{y}_i$ is the predicted label for the i^{th} testing sample and $\widehat{\sigma}$ is the estimated standard deviation of the noise term. Generally speaking, a smaller MSLL indicates a better testing prediction. We list the testing set MSLL for **GP**, **SE**, **SE+TC** and **GPST** in Table 5.2.

Model	$N = 200$	$N = 500$
GP	1.162±0.439	1.092±0.421
SE	2.457±1.898	2.999±3.167
SE+TC	0.972±0.214	0.919±0.123
GPST	$\mathbf{0.882 \pm 0.201}$	$\mathbf{0.835 \pm 0.156}$

Table 5.2: Test Mean Standardized Log Loss (MSLL) for the 4 variants of GP models. 95% confidence interval after $\pm$. Results are based on 10 repeated runs.

The result reveals that our method has statistically significantly smaller MSLL, as indicated by a one-sided paired t-test, compared to the other methods under both sample sizes. Also, the models with tensor contraction, including **SE+TC** and **GPST**, have smaller MSLL compared to their counterparts without tensor contraction, which further suggests that tensor contraction can be helpful for reducing the errors made by GP models with tensor data.

The estimators of the multi-linear kernel factors $\mathbf{K}_1, \mathbf{K}_2, \mathbf{K}_3$ and the feature maps of the Tensor-GPST model with $\lambda = 1.0$ for one random simulation dataset are visualized in Figure 5.3. One can see that the feature map $\widehat{\mathbf{W}}_{2,2}$ and $\widehat{\mathbf{W}}_{3,2}$ capture the corner and center blocks, and the covariances between the two feature maps are also high, as suggested by $\widehat{\mathbf{K}_1}(2,3) = 0.77$ and $\widehat{\mathbf{K}_2}(2,2) = 1.72$. Channels 1 & 3 have high covariances ($\widehat{\mathbf{K}_3}(1,3) = 3.5$), indicating that they share similar "corner signal" patterns and coincides with our ground truth setup (see Figure D.2a for the ground truth of $\mathbf{K}_3$).

Overall, the simulation experiments convey two messages:

- Adding the tensor contraction step leads to better regression performances when the signals have low-rank structures, robust to the choice of kernel, and the performance is similar to other low-rank tensor regression methods such as **CP** and **Tucker**.

- The anisotropic total variation penalty, though may not fully recover the underlying sparsity of the tensor contracting factors, can improve the regression performance of Tensor-GPST and also provides more direct interpretations.

The inferior performance of Tensor-GP (**GP**), however, is not suggesting that it is an inferior version of GP when dealing with tensor data, as we have demonstrated by comparing it against the GP with squared-exponential kernel (**SE**). The simulation pattern in Figure 5.2 contains randomness of the signal, making it more beneficial to extract features first using feature maps that cover multiple areas. Directly modeling the covariance structures among all pixels can be difficult in such scenarios.

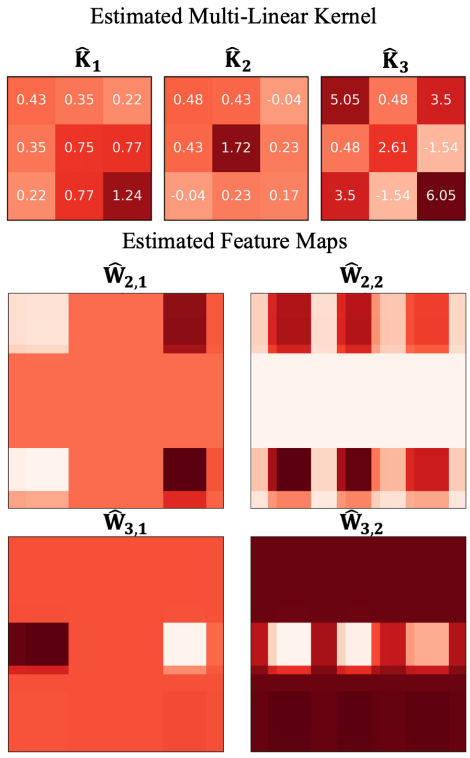


Figure 5.3: Estimated kernels (top) and non-zero feature maps (bottom) by **GPST** with $\lambda = 1.0$ for one random simulation dataset.

5.3.2 Application to Solar Flare Forecasting

A solar flare is an intense localized eruption of electromagnetic radiation in the Sun's atmosphere. Solar flares with high-energy radiation emission can strongly impact the Earth's space weather and potentially interfere the radio communication of the Earth. Recent works on solar flare forecasting (Bobra and Couvidat, 2015; Chen et al., 2019b; Wang et al., 2020; Jiao et al., 2020; Sun et al., 2022b) have demonstrated the effectiveness of using machine learning algorithms for forecasting flares, using either multivariate time-series data in the form of physical parameters or imaging data provided by the Solar Dynamics Observatory (SDO)/Helioseismic and Magnetic Imager (HMI) (Scherrer et al., 2012) and SDO/Atmospheric Imaging Assembly (AIA) (Lemen et al., 2012). It has been shown that these imaging data have low-dimensional representations that contain flare discriminating signals (Sun et al., 2021). Our methodology makes the astrophysical interpretation more accessible as compared to the previous deep learning approaches in that our model has a much shallower structure with only a feature extraction layer and a regression layer.

Here, we consider the specific problem of forecasting the intensity of a solar flare. In our dataset, we have $1,329$ flare samples from year 2010 to 2018, consisting of a total of

479 M-class and X-class flares and 850 B-class flares. The class of a flare is determined by the X-ray peak brightness in the range of 1-8 Å. The B-class flare has its brightness within $10^{-7} \sim 10^{-6}\text{W/m}^2$, which is considered weak and barely harmful, while the minimum M-class and X-class flares have brightness above 10^{-5} and 10^{-4}W/m^2, respectively. These more energetic flares are capable of heating and ionizing the upper atmosphere, resulting in brief radio blackouts and increased satellite drag. We collect the AIA-HMI imaging data for each flare, 1 hour prior to its peak, and each flare data is a 10-channel tensor data of size $50 \times 50 \times 10$, where spatial dimensions are binned down by roughly a factor of 10. We leave the data preprocessing steps and the astrophysical background to Appendix D.5.

Our goal here is to utilize the 10-channel tensor data $\boldsymbol{\mathcal{X}}_i$ to predict the flare intensity y_i and find the discriminating factors for M/X-class and B-class flares. We randomly split our dataset into a 75% training set (359 M/X/637 B) and a 25% testing set (120 M/X/213 B), and centering after log-transforming the flare intensity such that the B-class flare has $y_i \leq -0.5$ and M/X-class flare has $y_i \geq 0.5$.

We report the solar flare intensity prediction result across four different models: Tensor-GP (**GP**), Tensor-GPST (**GPST**), tensor regression with CP rank constraints (**CP**) and tensor regression with Tucker decomposition form (**Tucker**). The hyperparameters are set such that the models have the same latent dimensionality ($3 \times 3 \times 3$) or the rank (9 for **CP** and $3 \times 3 \times 3$ for **Tucker**) of the regression coefficients. The metrics used are rooted mean-squared error (RMSE), R-squared and MSLL. Additionally, we consider transforming the regression model to a binary classification model by thresholding the prediction at 0.0 such that any $\widehat{y}_i \geq 0$ indicates an M/X-class flare and any $\widehat{y}_i < 0$ indicates a B-class flare. Then we evaluate the resulting binary classification model with the True Skill Statistics (TSS)[1]. A skillful binary classifier for weak vs. strong solar flare is desirable for operational use. Results on the training and testing set are summarized in Table 5.3, with 10 random train/test splits.

Tensor-GP (**GP**) shows worse generalizability on the testing data as compared to the other three methods. **GPST** has slightly better testing set performance compared to **CP** and **Tucker**, but is not statistically significantly better than **Tucker**. Similar to the simulation data, the flare data exhibits randomness of the location of flare predictive signals, making the tensor contraction a critical step for improving the Tensor-GP method.

In Figure 5.4, we visualize the class-average AIA-131Å in the left column. There is a stark contrast between the two flare classes for this channel and many other channels as we show in Appendix D.6. A convenient output of our Tensor-GPST model is the direct

[1]True Skill Statistics is defined as: TSS = TP/(TP+FN) − FP/(FP+TN), where TP, TN, FP, FN represents true positive, true negative, false positive and false negative in the confusion matrix.

Model	Training (75% of the samples)			
	RMSE	R^2	MSLL	TSS
GP	0.646±0.019	0.336±0.044	1.028±0.134	0.466±0.039
CP	$\mathbf{0.564 \pm 0.035}$	$\mathbf{0.501 \pm 0.077}$	–	$\mathbf{0.625 \pm 0.069}$
Tucker	0.679±0.014	0.269±0.028	–	0.426±0.052
GPST	0.661±0.014	0.305± 0.023	1.005±0.021	0.449±0.040
Model	Testing (25% of the samples)			
	RMSE	R^2	MSLL	TSS
GP	0.772±0.239	0.182±0.114	1.138±0.085	0.362±0.159
CP	0.706±0.051	0.230±0.078	–	0.398±0.092
Tucker	0.683±0.040	0.259±0.079	–	0.414±0.134
GPST	0.681±0.043	0.265±0.087	$\mathbf{1.035 \pm 0.061}$	0.412±0.112

Table 5.3: Solar flare intensity regression performance on the training and testing sets for four tensor regression models. Results based on 10 random splits and 95% confidence intervals are provided after ±.

estimation of channel covariances in the multi-linear kernel, and we visualize the estimated $\widehat{\mathbf{K}}_3$ in the Figure as well. The estimated $\widehat{\mathbf{K}}_3$ reveals the important channel pairs when defining the similarity of pairs of tensor data, and we formalize this channel pair importance notion in (D.27) of Appendix D.6. To the best of our knowledge, our model is the first to consider the channel interactions for solar flare forecasting.

In the lower right panel of Figure 5.4, we visualize the pixels that have at least one feature map with weight $> 5 \times 10^{-3}$. These pixels contribute significantly to building the latent tensor, and are thus being considered as the most relevant pixels for solar flare prediction. As one can see, the selected pixels are concentrated around the two brightest spots of the AIA-131Å for the M/X-class and also around the boundary. These pixels contain two most significant flare discriminating factors: 1) the brightest spots of the AIA images; 2) the span of the bright regions (as M/X flares still have large AIA image intensities near the boundary but not B flares).

5.4 Conclusion

In this chapter, we propose a new methodology called Tensor-GPST for fitting Gaussian Process Regression (GPR) model on labelled multi-channel imaging data. We propose a tensor contraction operation to reduce the dimensionality of the tensor and also introduce anisotropic total variation penalty to the tensor contraction parameters to allow for interpretable feature extraction. We see improvements on the regression performances over the original Tensor-GP (Yu et al., 2018) in both simulation and the solar flare forecasting task.

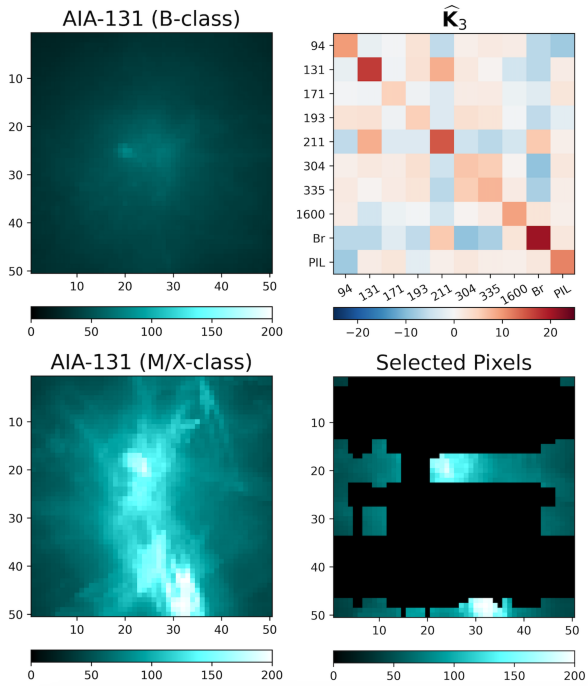


Figure 5.4: (Left column) The average AIA-131Å for all B-class flares and all M/X-class flares. (Right column) Estimated $\widehat{\mathbf{K}}_3$ in the multi-linear kernel that captures the channel-channel covariances (top). Pixels with at least one feature map with weight $> 5 \times 10^{-3}$ (bottom). We visualize the selected pixels with M/X class average AIA-131Å as the background. See full results in Appendix D.6.

The capability of the model in generating an interpretable low-dimensional tensor representation makes it ideal for many other scientific applications, such as predicting ADHD with Brain-Image (Li et al., 2018) and studying the association of brain connectivity with human traits (Papadogeorgou et al., 2021).

The current model has several limitations that can potentially lead to future research directions. First, we do not impose explicit identifiability constraints for the tensor contraction parameters and the multi-linear kernel parameters. This makes the optimization problem unconstrained thus enabling a simple gradient-based algorithm, but makes the parameters not fully identifiable. Second, the model has higher computational complexity as compared to the non-GP tensor regression models due to its GP formulation. A more efficient computational algorithm is needed to handle larger datasets.

Our code is available on GitHub at https://github.com/husun0822/TensorGPST.

CHAPTER 6

Concluding Remarks and Future Directions

In this thesis, we developed multiple novel statistical methodologies for tensor data with spatial, temporal, and data modality dimensions. Motivated by the applications in space weather monitoring, our proposed methods provide solutions to three major problems at the intersection of tensor data analysis and spatio-temporal statistics.

In Chapters 2 and 3, we develop a data-driven pipeline for the missing value imputation and uncertainty quantification of tensor data with spatially (and temporally) dependent data missingness. In Chapter 4, we undertake the forecasting problem for spatio-temporal tensor data with auxiliary non-spatial time series covariates and propose an autoregression framework that can merge the information from both spatial and non-spatial data modalities. In Chapter 5, we develop a Gaussian Process regression model with scalar response and multi-modal imaging covariate and propose a total-variation penalty for extracting predictive spatial signals from the covariates.

Each chapter provides new insights into modeling tensor data with dependencies and has systematic applications to space weather data including the geomagnetic index data and the solar flare data. These applications and the associated statistical methodologies could be of interest to domain scientists. Each chapter also points to several theoretical, methodological, and application future research directions for statisticians and practitioners, which we summarize below.

Hyperparameter Tuning of Spatio-Temporal Tensor Model Fitting models for spatio-temporal tensor data can be computationally intensive given the high dimensionality of the data. It is even worse that one needs to fine-tune the hyperparameters of the model, which can be impractical when the dimension of the hyperparameter is not low. Unfortunately, in each of the previous four chapters, our proposed method has at least two sets of hyperparameters, which we summarize in Table 6.1.

The tuning procedures we propose are often sub-optimal or only fine-tune a subset of the parameters. Given the high computational cost of conducting grid search, it is an urgent need for one to develop a computationally efficient tuning procedure for these tensor

Model	Hyperparameters	Tuning Procedure
VISTA (Chapter 2)	ℓ_2-regularization $(\lambda_1, \lambda_2, \lambda_3)$ spherical harmonics $(l_{\max}, \nu)$	Sequential Tuning
CTC (Chapter 3)	Tensor-train rank $\mathbf{r}$ Ising model likelihood $g(\cdot, \cdot)$	P-AIC for $\mathbf{r}$ Trial-and-error for $g(\cdot, \cdot)$
MARAC (Chapter 4)	lag P, Q functional norm penalty λ	BIC for P, Q cross-validation for λ
Tensor-GPST (Chapter 5)	latent tensor rank (r_1, r_2) TV-penalty λ	pre-determined r_1, r_2 cross-validation for λ

Table 6.1: Hyperparameters and the suggested tuning procedure in Chapters 2, 3, 4 and 5.

models or similar models. One potential approach is Bayesian optimization (Wu et al., 2019), whose empirical validity needs to be evaluated extensively on tensor data models.

Concentration Inequality of Locally-Dependent Tensor In Section 3.4 of Chapter 3, we only provide the estimator error bound and coverage guarantee of the conformal inference when the data are missing independently. The reason why we cannot derive similar results for the Ising model with locally-dependent missingness is that we cannot quantify the tail distribution of the spectral norm of the gradient tensor $\boldsymbol{\mathcal{G}}^*$.

For non-Gaussian likelihood models such as the binary tensor model considered in Chapter 3, the gradient tensor $\boldsymbol{\mathcal{G}}^*$ is non-zero but has zero-meaned sub-Gaussian entries. Concentration inequalities for such tensors with independent entries have been investigated in Tomioka and Suzuki (2014), as we restated in Lemma B.2.4. However, it is theoretically more challenging to generalize the results to dependent tensors.

Multi-Modal Tensor Inference In Chapter 4, we combine matrix-valued and vector-valued predictors in an autoregression framework for forecasting. The unique statistical challenge is to integrate the information of data from multiple sources with non-uniform tensor modes, and in Chapter 4, we consider the simplest setting of integrating matrices and vectors. More generally, one can consider integrating the information of tensors with arbitrary modes for joint statistical inference.

Existing works (Lock et al., 2013; Tang and Allen, 2021; Chen et al., 2021b) on analyzing multi-source data deal with data of uniform modes only. However, it is not uncommon to observe data with non-uniform modes. For example, in neuroscience studies, each patient has individual traits (vector), brain images (3-mode tensor), and electroencephalography (EEG) functional data of brain activity (matrix). New methods on how to extract the shared and individual representations of these tensor data can be significantly beneficial for these applications.

Transformation-Invariant Spatial Tensor Learning In Chapter 5, we consider the

scalar-on-tensor regression task and apply the model to a solar flare forecasting dataset. The solar flare data is carefully pre-processed before the model fitting because each flare happens in a different active region and the imaging data can have different sizes for different active regions. Even worse, the solar flare can happen at random locations in the image and the flare precursors, which are typically a bright band in the image, can have random shapes.

Modern deep learning models such as the convolutional neural network (CNN) (LeCun et al., 2015; Krizhevsky et al., 2017) can accommodate these artifacts in data as the convolution operation can extract features in a transformation-invariant fashion. However, for tensor regression models such as tensor linear regression or tensor Gaussian Process regression, the tensor data needs to be calibrated in the preprocessing steps to have a standardized grid before fitting.

The calibration of tensor data with manual preprocessing, however, can be cumbersome, as we have demonstrated for the solar flare data in Appendix D.5. A more favorable approach is to make the tensor regression model to be insensitive to un-standardized data. There are two potential research tracks to make this possible.

The first track is to estimate the transformation parameters of the tensor data together with the regression parameters. The spatial transformer network (Jaderberg et al., 2015) follows this idea and learns the affine transformation parameter of the data in the first few layers of a deep neural network and then propagates the transformed data to the subsequent layers for classification or regression. It would be interesting to develop a statistical method for imaging data that can accomplish this in a more interpretable manner.

The second track is to develop a transformation-invariant regression model. In van der Wilk et al. (2018), a Gaussian Process regression model is formulated via a transformation-invariant kernel. This new kernel captures the covariances of two samples via the sum of the covariances of all possible pairs of transformed samples. However, this approach requires that the possible transformations of samples are finite. An interesting direction to pursue is to find an alternative transformation-invariant representation of the tensor covariates/coefficients, e.g., a frequency-domain representation via low tubal-rank tensor decomposition (Wang et al., 2021a; Roy and Michailidis, 2022).

If one can construct such a framework that enables transformation-invariant tensor learning, it will benefit applications such as the solar flare forecasting task as one can utilize the full-sized solar imaging data without manual pre-processing. However, it might come with extra computational costs for transformation-invariant tensor learning. Modern computer vision models with attention mechanism (Jaderberg et al., 2015) will be easier to implement at the moment if one needs to deploy the model for online forecast-

ing.

Tensor Quantile Model Throughout the thesis, we focus on the modeling of the conditional mean of the quantity of interest given the tensor covariates, such as the missing values of tensors or the future matrix data that we want to forecast. But oftentimes, it is also of scientific interest to quantify the tail behavior. For example, for the scalar-on-tensor regression model discussed in Chapter 5, we forecast the mean solar flare intensity conditioning on the multi-channel imaging data. For astrophysicists, the only concerning solar flares are the extremely strong events. Therefore, it is more informative to predict the super-quantiles of the scalar response given tensor covariates.

There are several recent works (Lu et al., 2020; Li and Zhang, 2021; Liu et al., 2024) on scalar-on-tensor quantile regressions or tensor-on-scalar quantile regressions (Wei et al., 2023). Quantile, however, is not a summary of the entire tail of the response distribution. The expected shortfall, defined as the conditional mean of the response in the upper/lower tail of the distribution, is a more thorough metric for summarizing the tail behavior. An interesting research direction is to investigate the expected shortfall regression with tensor covariates or response, which generalizes the existing framework under the linear regression setup (Dimitriadis and Bayer, 2019; Barendse, 2022; He et al., 2023; Zhang et al., 2023).

Online Spatio-Temporal Tensor Learning Spatio-temporal data that we investigated in this thesis is an offline dataset, where all data are available at once and need to be loaded together for model fitting and inference. In an operational setting, however, spatio-temporal data arrives in streams. Transforming the model estimation procedure from offline to online (Cai et al., 2023) can significantly reduce the storage demand and get updated models in a timely fashion. It is an interesting research direction to develop an online spatio-temporal tensor learning framework that can make forecasts or impute missing values using the latest arrived stream data. It is also interesting to devise statistical methodologies that can let the model unlearn (Cao and Yang, 2015) the earliest data since perhaps only the latest spatial data are relevant for tensor completion and regression for the next time point.

APPENDIX A

Appendix for Chapter 2

A.1 Proof of Theorem 2.2.1

Proof. The objective function $F(\mathbf{A}_{1:T}^{(k)}, \mathbf{B}_{1:T}^{(k)})$ has the property:

$$\begin{aligned} F(\mathbf{A}_{1:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}) &= \widetilde{Q}(\mathbf{A}_1^{(k)} | \mathbf{A}_{1:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}) \\ &\geq \inf_{\mathbf{A}_1} \widetilde{Q}(\mathbf{A}_1 | \mathbf{A}_{1:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}) \\ &= \widetilde{Q}(\mathbf{A}_1^{(k+1)} | \mathbf{A}_{1:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}) \qquad \text{(A.1)} \\ &\geq F(\mathbf{A}_1^{(k+1)}, \mathbf{A}_{2:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}), \qquad \text{(A.2)} \end{aligned}$$

where the definition of $\widetilde{Q}$ is in (2.7). Equation (A.1) holds because we update $\mathbf{A}_1$ to be $\mathbf{A}_1^{(k+1)}$ using ridge regression: $\mathbf{A}_1^{(k+1)} = \arg\min_{\mathbf{A}_1} \widetilde{Q}(\mathbf{A}_1 | \mathbf{A}_{1:T}^{(k)}, \mathbf{B}_{1:T}^{(k)})$. Inequality (A.2) holds because $\widetilde{Q}(\mathbf{A}_1^{(k+1)} | \mathbf{A}_{1:T}^{(k)}, \mathbf{B}_{1:T}^{(k)})$ is the upper bound of $F(\mathbf{A}_1^{(k+1)}, \mathbf{A}_{2:T}^{(k)}, \mathbf{B}_{1:T}^{(k)})$, as we majorize the first term of the objective function using inequality (2.6).

The property above indicates that after one single update of matrix $\mathbf{A}_1$, the value of the objective function is non-increasing. Applying a similar argument for all other matrices $\mathbf{A}_2, \mathbf{A}_3, \ldots, \mathbf{A}_T, \mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_T$ leads to a chain of inequalities:

$$\begin{aligned} F(\mathbf{A}_{1:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}) \geq F(\mathbf{A}_1^{(k+1)}, \mathbf{A}_{2:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}) \geq F(\mathbf{A}_{1:2}^{(k+1)}, \mathbf{A}_{3:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}) \geq \cdots \geq F(\mathbf{A}_{1:T}^{(k+1)}, \mathbf{B}_{1:T}^{(k)}) \\ \geq F(\mathbf{A}_{1:T}^{(k+1)}, \mathbf{B}_1^{(k)}, \mathbf{B}_{2:T}^{(k)}) \geq F(\mathbf{A}_{1:T}^{(k+1)}, \mathbf{B}_{1:2}^{(k)}, \mathbf{B}_{3:T}^{(k)}) \geq \ldots F(\mathbf{A}_{1:T}^{(k+1)}, \mathbf{B}_{1:T}^{(k+1)}), \end{aligned}$$

which proves that each update of $\mathbf{A}_t$ or $\mathbf{B}_t$ goes towards a descent direction. □

A.2 Proof of Theorem 2.2.2

Proof. Note that in Appendix A.1, we proved inequality (A.2). More generally, for any arbitrary t, we have the following:

$$\begin{aligned}\Delta_{k,t}^{\mathbf{A}} \triangleq & F(\mathbf{A}_{1:t-1}^{(k+1)}, \mathbf{A}_{t:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}) - F(\mathbf{A}_{1:t}^{(k+1)}, \mathbf{A}_{t+1:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}) \\ \geq & \widetilde{Q}(\mathbf{A}_t^{(k)} | \mathbf{A}_{1:t-1}^{(k+1)}, \mathbf{A}_{t:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}) - \widetilde{Q}(\mathbf{A}_t^{(k+1)} | \mathbf{A}_{1:t-1}^{(k+1)}, \mathbf{A}_{t:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}). \end{aligned} \tag{A.3}$$

The right hand side of (A.3) is the difference of $\widetilde{Q}(\mathbf{A}_t | \mathbf{A}_{1:t-1}^{(k+1)}, \mathbf{A}_{t:T}^{(k)}, \mathbf{B}_{1:T}^{(k)})$ evaluated at $\mathbf{A}_t^{(k)}$ and $\mathbf{A}_t^{(k+1)}$. Recall that:

$$\begin{aligned}\widetilde{Q}(\mathbf{A}_t | \mathbf{A}_{1:t-1}^{(k+1)}, \mathbf{A}_{t:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}) \triangleq & \frac{1}{2}\|\mathbf{X}_t^{(k)} - \mathbf{A}_t(\mathbf{B}_t^{(k)})^\top\|_{\mathrm{F}}^2 + \frac{\lambda_1}{2}\|\mathbf{A}_t\|_{\mathrm{F}}^2 + \frac{\lambda_3}{2}\|\mathbf{Y}_t - \mathbf{A}_t(\mathbf{B}_t^{(k)})^\top\|_{\mathrm{F}}^2 \\ & + \frac{\lambda_2}{2}\mathbf{I}_{\{t>1\}}\|\mathbf{A}_t(\mathbf{B}_t^{(k)})^\top - \mathbf{A}_{t-1}^{(k+1)}(\mathbf{B}_{t-1}^{(k)})^\top\|_{\mathrm{F}}^2 \\ & + \frac{\lambda_2}{2}\mathbf{I}_{\{t<T\}}\|\mathbf{A}_{t+1}^{(k)}(\mathbf{B}_{t+1}^{(k)})^\top - \mathbf{A}_t(\mathbf{B}_t^{(k)})^\top\|_{\mathrm{F}}^2.\end{aligned}$$

Note that this is a quadratic form of $\mathbf{A}_t$ thus higher order (≥ 3) derivatives are all zero. We can do a Taylor expansion for $\widetilde{Q}(\mathbf{A}_t^{(k)} | \mathbf{A}_{1:t-1}^{(k+1)}, \mathbf{A}_{t:T}^{(k)}, \mathbf{B}_{1:T}^{(k)})$ at $\mathbf{A}_t^{(k+1)}$:

$$\begin{aligned}\widetilde{Q}(\mathbf{A}_t^{(k)} | \mathbf{A}_{1:t-1}^{(k+1)}, \mathbf{A}_{t:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}) = & \widetilde{Q}(\mathbf{A}_t^{(k+1)} | \mathbf{A}_{1:t-1}^{(k+1)}, \mathbf{A}_{t:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}) \\ & + (\nabla\widetilde{Q})(\mathbf{A}_t^{(k)} - \mathbf{A}_t^{(k+1)}) \\ & + \frac{1}{2}(\mathbf{A}_t^{(k)} - \mathbf{A}_t^{(k+1)})^\top\mathbf{H}(\mathbf{A}_t^{(k)} - \mathbf{A}_t^{(k+1)}),\end{aligned} \tag{A.4}$$

where $\mathbf{H} = (1 + \lambda_2(1 + \mathbf{I}_{\{2\leq t\leq T-1\}}) + \lambda_3)(\mathbf{B}_t^{(k)})^\top\mathbf{B}_t^{(k)} + \lambda_1\mathbf{I}$. We have $\nabla\widetilde{Q} = 0$ since $\mathbf{A}_t^{(k+1)}$ is the minimizer of $\widetilde{Q}(\mathbf{A}_t | \mathbf{A}_{1:t-1}^{(k+1)}, \mathbf{A}_{t:T}^{(k)}, \mathbf{B}_{1:T}^{(k)})$. Combining (A.3) and (A.4), one can see that:

$$\begin{aligned}\Delta_{k,t}^{\mathbf{A}} \geq & \widetilde{Q}(\mathbf{A}_t^{(k)} | \mathbf{A}_{1:t-1}^{(k+1)}, \mathbf{A}_{t:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}) - \widetilde{Q}(\mathbf{A}_t^{(k+1)} | \mathbf{A}_{1:t-1}^{(k+1)}, \mathbf{A}_{t:T}^{(k)}, \mathbf{B}_{1:T}^{(k)}) \\ = & \frac{1}{2}(\mathbf{A}_t^{(k)} - \mathbf{A}_t^{(k+1)})^\top\mathbf{H}(\mathbf{A}_t^{(k)} - \mathbf{A}_t^{(k+1)}) \\ = & \frac{1 + \lambda_2(1 + \mathbf{I}_{\{2\leq t\leq T-1\}}) + \lambda_3}{2}\|(\mathbf{A}_t^{(k)} - \mathbf{A}_t^{(k+1)})(\mathbf{B}_t^{(k)})^\top\|_{\mathrm{F}}^2 \\ & + \frac{\lambda_1}{2}\|\mathbf{A}_t^{(k)} - \mathbf{A}_t^{(k+1)}\|_{\mathrm{F}}^2.\end{aligned} \tag{A.5}$$

Similarly for any updates of $\mathbf{B}_t$, we have:

$$\begin{aligned}\Delta_{k,t}^{\mathbf{B}} &\triangleq F(\mathbf{A}_{1:T}^{(k+1)}, \mathbf{B}_{1:t-1}^{(k+1)}, \mathbf{B}_{t:T}^{(k)}) - F(\mathbf{A}_{1:T}^{(k+1)}, \mathbf{B}_{1:t}^{(k+1)}, \mathbf{B}_{t+1:T}^{(k)}) \\ &\geq \widetilde{Q}(\mathbf{B}_t^{(k)} | \mathbf{A}_{1:T}^{(k+1)}, \mathbf{B}_{1:t-1}^{(k+1)}, \mathbf{B}_{t:T}^{(k)}) - \widetilde{Q}(\mathbf{B}_t^{(k+1)} | \mathbf{A}_{1:T}^{(k+1)}, \mathbf{B}_{1:t-1}^{(k+1)}, \mathbf{B}_{t:T}^{(k)}) \\ &= \frac{1 + \lambda_2(1 + \mathbf{I}_{\{2 \leq t \leq T-1\}}) + \lambda_3}{2} \|\mathbf{A}_t^{(k+1)}(\mathbf{B}_t^{(k)} - \mathbf{B}_t^{(k+1)})^\top\|_{\mathrm{F}}^2 \\ &+ \frac{\lambda_1}{2}\|\mathbf{B}_t^{(k)} - \mathbf{B}_t^{(k+1)}\|_{\mathrm{F}}^2. \end{aligned} \tag{A.6}$$

Since (A.5) and (A.6) hold for all $\mathbf{A}_t$ and $\mathbf{B}_t$, we can take the sum of $\Delta_{k,t}^{\mathbf{A}}, \Delta_{k,t}^{\mathbf{B}}$ across all t. Note that $\sum_t(\Delta_{k,t}^{\mathbf{A}} + \Delta_{k,t}^{\mathbf{B}}) = \Delta_k$. The right-hand side is the lower bound for Δ_k in (2.16). □

A.3 Proof of Theorem 2.2.3

Proof. The first result can be easily proved by noting that

$$F(\mathbf{A}_{1:T}^{(1)}, \mathbf{B}_{1:T}^{(1)}) - f^\infty \geq F(\mathbf{A}_{1:T}^{(1)}, \mathbf{B}_{1:T}^{(1)}) - F(\mathbf{A}_{1:T}^{(K)}, \mathbf{B}_{1:T}^{(K)}) = \sum_{k=1}^{K} \Delta_k \geq K(\min_{1 \leq k \leq K} \Delta_k). \tag{A.7}$$

Given the assumption that $l^L\mathbf{I} \leq (\mathbf{A}_t^{(k)})^\top \mathbf{A}_t^{(k)} \leq l^U\mathbf{I}$, $l^L\mathbf{I} \leq (\mathbf{B}_t^{(k)})^\top \mathbf{B}_t^{(k)} \leq l^U\mathbf{I}$ for all t, k. Equations (2.18) and (2.19) can be proved with the following inequalities:

$$l^L\|\mathbf{A}_t^{(k)} - \mathbf{A}_t^{(k+1)}\|_{\mathrm{F}}^2 \leq \|(\mathbf{A}_t^{(k)} - \mathbf{A}_t^{(k+1)})(\mathbf{B}_t^{(k)})^\top\|_{\mathrm{F}}^2 \leq l^U\|\mathbf{A}_t^{(k)} - \mathbf{A}_t^{(k+1)}\|_{\mathrm{F}}^2; \tag{A.8}$$

$$l^L\|\mathbf{B}_t^{(k)} - \mathbf{B}_t^{(k+1)}\|_{\mathrm{F}}^2 \leq \|\mathbf{A}_t^{(k+1)}(\mathbf{B}_t^{(k)} - \mathbf{B}_t^{(k+1)})^\top\|_{\mathrm{F}}^2 \leq l^U\|\mathbf{B}_t^{(k)} - \mathbf{B}_t^{(k+1)}\|_{\mathrm{F}}^2. \tag{A.9}$$

Given the lower bound in Theorem 2.2.2 and the inequality in (A.7), we have:

$$\begin{aligned}&\frac{F(\mathbf{A}_{1:T}^{(1)}, \mathbf{B}_{1:T}^{(1)}) - f^\infty}{K} \geq \min_{1 \leq k \leq K} \Delta_k \\ &\geq \min_{1 \leq k \leq K} \left\{ \frac{\lambda_1}{2} \sum_{t=1}^{T} \left(\|\mathbf{A}_t^{(k)} - \mathbf{A}_t^{(k+1)}\|_{\mathrm{F}}^2 + \|\mathbf{B}_t^{(k)} - \mathbf{B}_t^{(k+1)}\|_{\mathrm{F}}^2 \right) \right. \\ &\left. + \frac{1}{2} \sum_{t=1}^{T} (1 + \lambda_2 + \lambda_3) \left(\|(\mathbf{A}_t^{(k)} - \mathbf{A}_t^{(k+1)})(\mathbf{B}_t^{(k)})^\top\|_{\mathrm{F}}^2 + \|\mathbf{A}_t^{(k+1)}(\mathbf{B}_t^{(k)} - \mathbf{B}_t^{(k+1)})^\top\|_{\mathrm{F}}^2 \right) \right\} \\ &\geq \min_{1 \leq k \leq K} \left\{ \frac{l^L(1 + \lambda_2 + \lambda_3) + \lambda_1}{2} \sum_{t=1}^{T} \left(\|\mathbf{A}_t^{(k)} - \mathbf{A}_t^{(k+1)}\|_{\mathrm{F}}^2 + \|\mathbf{B}_t^{(k)} - \mathbf{B}_t^{(k+1)}\|_{\mathrm{F}}^2 \right) \right\}. \end{aligned}$$

The last step uses the inequalities on the LHS in (A.8) and (A.9). This proves (2.18). Using the inequality on the RHS in (A.8) and (A.9) yields (2.19). □

APPENDIX B

Appendix for Chapter 3

B.1 Proofs of Theorems and Propositions

Throughout this section, for any tensor $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$, we use $\bar{d}, d^*$ to denote $\sum_k d_k$ and $\prod_k d_k$, respectively. For any tensor-train rank $\mathbf{r} = (r_1, \ldots, r_{K-1})$, we use r^* to denote $\prod_k r_k$. We use $c, c', C, C_0, C_1, \ldots$ to denote positive absolute constants and $c_K, c'_K, C_K, C_{K,0}, C_{K,1}, \ldots$ to denote positive constants that only relate to K. For two sequences $\{a_n\}_{n=1}^{\infty}$ and $\{b_n\}_{n=1}^{\infty}$, we use $a_n \asymp b_n$ to represent $\lim_{n\to\infty} a_n/b_n = C > 0$, with C being finite.

B.1.1 Proof of Proposition 3.2.1

Proof. Given any testing entry $s^* \in \mathbb{S}_{miss}$, we relabel all elements in $\mathbb{S}_{cal} \cup \{s^*\}$ as $\{s_1, \ldots, s_{n+1}\}$. Now recall the definition of $\boldsymbol{\mathcal{E}}_0$ as:

$$\boldsymbol{\mathcal{E}}_0 = \left\{ \widetilde{\boldsymbol{\mathcal{W}}}_s = 1 \text{ for } s \in \mathbb{S}_{tr} \cup \mathbb{S}_{cal}, \mathbb{S}_{cal} \cup \{s^*\} = \{s_1, \ldots, s_{n+1}\} \text{ and } \widetilde{\boldsymbol{\mathcal{W}}}_s = -1 \text{ o.w.} \right\},$$

namely one observes data only at $\mathbb{S}_{tr}$ and n out of $n+1$ entries from $\{s_1, \ldots, s_{n+1}\}$.

Let V denote the non-conformity score of the testing entry, then the weighted exchangeability framework in Tibshirani et al. (2019) states that one can treat V as a weighted draw from $\{\mathcal{S}(\boldsymbol{\mathcal{X}}_{s_1}, \widehat{\boldsymbol{\mathcal{X}}}_{s_1}), \ldots, \mathcal{S}(\boldsymbol{\mathcal{X}}_{s_{n+1}}, \widehat{\boldsymbol{\mathcal{X}}}_{s_{n+1}})\}$, with weight being:

$$\mathrm{P}\left[V = \mathcal{S}(\boldsymbol{\mathcal{X}}_{s_k}, \widehat{\boldsymbol{\mathcal{X}}}_{s_k}) \middle| \boldsymbol{\mathcal{E}}_0 \right] = \frac{\mathrm{P}\left[\widetilde{\boldsymbol{\mathcal{W}}}_{s_k} = -1, \widetilde{\boldsymbol{\mathcal{W}}}_s = 1 \text{ for } s \in \mathbb{S}_k \middle| \boldsymbol{\mathcal{E}}_0 \right]}{\sum_{l=1}^{n+1} \mathrm{P}\left[\widetilde{\boldsymbol{\mathcal{W}}}_{s_l} = -1, \widetilde{\boldsymbol{\mathcal{W}}}_s = 1 \text{ for } s \in \mathbb{S}_l \middle| \boldsymbol{\mathcal{E}}_0 \right]},$$

where $\mathbb{S}_k = \{s_1, \ldots, s_{n+1}\} \setminus \{s_k\}$, for $k = 1, \ldots, n+1$. Multiplying both the numerator and the denominator by $\mathrm{P}(\boldsymbol{\mathcal{E}}_0)$ leads to the weight in the form of $p_k / \sum_{l=1}^{n+1} p_l$, with p_k defined

as (3.4). The coverage guarantee in (3.6) is then a direct result of Theorem 2 of Tibshirani et al. (2019). □

B.1.2 Proof of Theorem 3.4.3

Proof. Using Taylor expansion upon $\ell(\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}|\widehat{\boldsymbol{\mathcal{B}}})$ at $\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{B}}^*$ yields:

$$\ell(\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}|\widehat{\boldsymbol{\mathcal{B}}}) = \ell(\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}|\boldsymbol{\mathcal{B}}^*) + \left\langle \nabla\ell(\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}|\boldsymbol{\mathcal{B}}^*), \widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^* \right\rangle + \frac{1}{2}\mathbf{vec}(\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*)^\top \mathbf{H}(\check{\boldsymbol{\mathcal{B}}})\mathbf{vec}(\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*), \tag{B.1}$$

where $\check{\boldsymbol{\mathcal{B}}}$ is a convex combination of $\widehat{\boldsymbol{\mathcal{B}}}$ and $\boldsymbol{\mathcal{B}}^*$. Since, by assumption, $\widehat{\boldsymbol{\mathcal{B}}}$ reaches the global minimum of $\ell(\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}|\boldsymbol{\mathcal{B}})$, or $\ell(\boldsymbol{\mathcal{B}})$ in short, we have $\ell(\widehat{\boldsymbol{\mathcal{B}}}) \leq \ell(\boldsymbol{\mathcal{B}}^*)$, and thus the sum of the last two terms in (B.1) are no greater than zero.

For the first term, let $\boldsymbol{\mathcal{G}}^* = \nabla\ell(\boldsymbol{\mathcal{B}}^*)$ and $\boldsymbol{\mathcal{G}}^*$ satisfies:

$$[\boldsymbol{\mathcal{G}}^*]_s = -[1 - f(\boldsymbol{\mathcal{B}}_s^*)] \cdot 2h'(x) \cdot \mathbb{1}_{\{[\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}]_s = 1\}} + \frac{qf(\boldsymbol{\mathcal{B}}_s^*)[1 - f(\boldsymbol{\mathcal{B}}_s^*)]}{1 - qf(\boldsymbol{\mathcal{B}}_s^*)} \cdot 2h'(x) \cdot \mathbb{1}_{\{[\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}]_s = -1\}}, \tag{B.2}$$

and it is easy to verify that $\mathrm{E}[[\boldsymbol{\mathcal{G}}^*]_s] = 0$ and $\|\boldsymbol{\mathcal{G}}^*\|_\infty \leq \alpha_\xi$. By Lemma B.2.1, we can lower bound the first term as:

$$\left\langle \nabla\ell(\boldsymbol{\mathcal{W}}_{\mathbb{S}_{tr}}|\boldsymbol{\mathcal{B}}^*), \widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^* \right\rangle \geq -\|\boldsymbol{\mathcal{G}}^*\|_\sigma \|\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_*. \tag{B.3}$$

By Lemma B.2.2, we have $\mathrm{rank}^{\mathrm{tt}}(\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*) \leq 2\mathbf{r}$, and then by Lemma B.2.3, we have $\|\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_* \leq \sqrt{(2r_1)\cdots(2r_{K-1})} \cdot \|\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_{\mathrm{F}}$. Therefore, to lower bound the RHS of (B.3), we only need to upper bound the spectral norm of $\boldsymbol{\mathcal{G}}^*$. Since entry-wisely, $\boldsymbol{\mathcal{G}}^*$ is mean-zero and bounded by α_ξ (therefore the sub-Gaussian norm is α_ξ), we can apply Lemma B.2.4 and get:

$$\mathrm{P}\left(\|\boldsymbol{\mathcal{G}}^*\|_\sigma \leq \sqrt{8\alpha_\xi^2\left[\bar{d}\log 5K + \log\frac{2}{\delta}\right]}\right) \geq 1 - \delta. \tag{B.4}$$

By setting $\delta = \exp\left(-C_1\bar{d}\log K\right)$, with C_1 be some absolute constant, we can simplify (B.4) as:

$$\mathrm{P}\left(\|\boldsymbol{\mathcal{G}}^*\|_\sigma \leq C_K\alpha_\xi\sqrt{\bar{d}}\right) \geq 1 - \exp\left(-C_1\bar{d}\log K\right), \tag{B.5}$$

with $C_K = \sqrt{8\left(\log 5K + C_1\log K + 1\right)}$.

Combining these results, we can lower bound the RHS of (B.3) by:

$$-\|\boldsymbol{\mathcal{G}}^*\|_\sigma\|\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_* \geq -C_{K,1}\alpha_\xi\sqrt{\bar{d}r^*}\|\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_{\mathrm{F}}, \tag{B.6}$$

with probability at least $1-\exp\left(-C_1\bar{d}\log K\right)$, where $C_{K,1}=2^{(K-1)/2}C_K$.

For the quadratic form in (B.1), we have:

$$\frac{1}{2}\mathbf{vec}(\widehat{\boldsymbol{\mathcal{B}}}-\boldsymbol{\mathcal{B}}^*)^\top\mathbf{H}(\check{\boldsymbol{\mathcal{B}}})\mathbf{vec}(\widehat{\boldsymbol{\mathcal{B}}}-\boldsymbol{\mathcal{B}}^*)\geq\frac{\gamma_\xi}{2}\|\widehat{\boldsymbol{\mathcal{B}}}-\boldsymbol{\mathcal{B}}^*\|_{\mathrm{F}}^2>0. \tag{B.7}$$

Combining (B.6) and (B.7), we obtain:

$$\mathrm{P}\left(\frac{1}{\sqrt{d^*}}\|\widehat{\boldsymbol{\mathcal{B}}}-\boldsymbol{\mathcal{B}}^*\|_{\mathrm{F}}\leq 2C_{K,1}\frac{\alpha_\xi}{\gamma_\xi}\sqrt{\frac{r^*\bar{d}}{d^*}}\right)\geq 1-\exp\left(-C_1\bar{d}\log K\right),$$

which completes the proof. □

B.1.3 Proof of Theorem 3.4.5

To begin with, we state an essential lemma, which is a trivial extension of Theorem 3.2 of (Gui et al., 2023) under the conformalized matrix completion context:

Lemma B.1.1 (Theorem 3.2 of (Gui et al., 2023)). *Let $\widehat{\boldsymbol{\mathcal{X}}}$ be the output of any tensor completion algorithm, and $\widehat{\boldsymbol{\mathcal{B}}}$ be the output of the RGrad algorithm and both $\widehat{\boldsymbol{\mathcal{X}}},\widehat{\boldsymbol{\mathcal{B}}}$ are based on $\mathbb{S}_{tr}$ only, then given that $g(x,y)=0$, we have:*

$$\mathrm{E}\left[\frac{1}{|\mathbb{S}_{miss}|}\sum_{s\in\mathbb{S}_{miss}}\mathbb{1}_{\{\boldsymbol{\mathcal{X}}_s\in\widehat{C}_{1-\alpha,s}(\widehat{\boldsymbol{\mathcal{X}}})\}}\right]\geq 1-\alpha-\mathrm{E}[\Delta], \tag{B.8}$$

where $\widehat{C}_{1-\alpha,s}(\widehat{\boldsymbol{\mathcal{X}}})$ is the conformal interval for testing entry s at $(1-\alpha)$ level by the CTC algorithm and Δ is defined as:

$$\Delta=\frac{1}{2}\sum_{s\in\mathbb{S}_{cal}\cup\{s^*\}}\left|\frac{\exp[-2h(\widehat{\boldsymbol{\mathcal{B}}}_s)]}{\sum_{s\in\mathbb{S}_{cal}\cup\{s^*\}}\exp[-2h(\widehat{\boldsymbol{\mathcal{B}}}_s)]}-\frac{\exp[-2h(\boldsymbol{\mathcal{B}}_s^*)]}{\sum_{s\in\mathbb{S}_{cal}\cup\{s^*\}}\exp[-2h(\boldsymbol{\mathcal{B}}_s^*)]}\right|. \tag{B.9}$$

We neglect the proof here since the generalization from matrix to tensor setting is trivial as one can matricize the tensor into a matrix and the result holds automatically. By Lemma A.1 in (Gui et al., 2023), one can further upper bound Δ by:

$$\Delta\leq\frac{\|\exp[-2h(\widehat{\boldsymbol{\mathcal{B}}})]-\exp[-2h(\boldsymbol{\mathcal{B}}^*)]\|_1}{\sum_{s\in\mathbb{S}_{cal}}\exp[-2h(\widehat{\boldsymbol{\mathcal{B}}}_s)]}, \tag{B.10}$$

where the $h(\cdot)$ is applied to tensors element-wisely and $\|\cdot\|_1$ is the element-wise tensor ℓ_1 norm. The quantity Δ is trivially bounded by 1 as it is the total-variation (TV) distance

between two CDFs of discrete random variables. With this lemma, we now formally prove Theorem 3.4.5.

Proof. Given Lemma B.1.1, the coverage guarantee can be derived if one can characterize an upper bound for $\mathrm{E}[\Delta]$. To upper bound Δ, we start from (B.10) and bound the numerator on the RHS of (B.10) as:

$$\begin{aligned}\|\exp[-2h(\widehat{\boldsymbol{\mathcal{B}}})] - \exp[-2h(\boldsymbol{\mathcal{B}}^*)]\|_1 &\leq \sup_{|x|\leq\xi} |\exp[-2h(x)] \cdot 2h'(x)| \cdot \|\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_1 \\ &\leq u_\xi \alpha_\xi \cdot \|\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_1 \leq u_\xi \alpha_\xi \cdot \sqrt{d^*} \|\widehat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}^*\|_{\mathrm{F}}. \end{aligned} \tag{B.11}$$

Then we can apply the result of Theorem 3.4.3 to further bound (B.11) with high probability.

For the denominator of the RHS of (B.10), we can lower bound it first as $n_{cal} l_\xi$, and for n_{cal}, since each tensor entry can become a calibration point independently with probability $\exp[-2h(\boldsymbol{\mathcal{B}}_s)](1-q)$, where $0 < q < 1$ is the train-calibration set split probability, we can then apply the Chernoff bound and obtain:

$$\mathrm{P}\left(n_{cal} \leq (1-c)(1-q)\|\exp[-2h(\boldsymbol{\mathcal{B}}^*)]\|_1\right) \leq \exp\left[-\frac{c^2(1-q)\|\exp[-2h(\boldsymbol{\mathcal{B}}^*)]\|_1}{2}\right], \tag{B.12}$$

for any $0 < c < 1$. By denoting the event $\{n_{cal} \geq (1-c)(1-q)\|\exp[-2h(\boldsymbol{\mathcal{B}}^*)]\|_1\}$ as $\mathcal{E}_0$ and the event in (3.23) as $\mathcal{E}_1$ and noticing that $\|\exp[-2h(\boldsymbol{\mathcal{B}}^*)]\|_1 \geq d^* l_\xi$, then we have:

$$\mathrm{P}\left(\Delta \leq \frac{2C_{K,1}}{(1-c)(1-q)} \cdot \frac{u_\xi \alpha_\xi^2}{\gamma_\xi l_\xi^2} \cdot \sqrt{\frac{r^* \bar{d}}{d^*}}\right) \geq 1 - \exp[-C_1 \bar{d} \log K] - \exp\left[-\frac{c^2(1-q)d^* l_\xi}{2}\right],$$

where the probability is the lower bound of the probability of the event $\mathcal{E}_0 \cap \mathcal{E}_1$. With this tail bound on Δ, one can upper bound $\mathrm{E}[\Delta]$ as:

$$\mathrm{E}[\Delta] \leq \frac{2C_{K,1}}{(1-c)(1-q)} \cdot \frac{u_\xi \alpha_\xi^2}{\gamma_\xi l_\xi^2} \cdot \sqrt{\frac{r^* \bar{d}}{d^*}} + \exp[-C_1 \bar{d} \log K] + \exp\left[-\frac{c^2(1-q)d^* l_\xi}{2}\right], \tag{B.13}$$

and thereby completes the proof. □

B.2 Technical Lemmas

All technical lemmas listed in this section are cited from existing works. Therefore, we omit the proof here and refer our readers to the corresponding papers cited.

Lemma B.2.1 (Lemma 1 of Wang and Li (2020)). *For two tensors $\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$, their inner product $\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle$ can be bounded as:*

$$|\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle| \leq \|\boldsymbol{\mathcal{A}}\|_\sigma \|\boldsymbol{\mathcal{B}}\|_*,$$

where $\|\cdot\|_\sigma, \|\cdot\|_*$ are the tensor spectral norm and the tensor nuclear norm, respectively.

Lemma B.2.2 (Lemma 24 of Cai et al. (2022d)). *Let $\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$ be two low tensor-train rank tensors with $\mathrm{rank}^{tt}(\boldsymbol{\mathcal{A}}) = \mathbf{r}_1, \mathrm{rank}^{tt}(\boldsymbol{\mathcal{B}}) \leq \mathbf{r}_2$, respectively. Then one has:*

$$\mathrm{rank}^{tt}(\boldsymbol{\mathcal{A}} + \boldsymbol{\mathcal{B}}) \leq \mathbf{r}_1 + \mathbf{r}_2.$$

Lemma B.2.3 (Lemma 25 of Cai et al. (2022d)). *Let $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$ be a low tensor-train rank tensor with $\mathrm{rank}^{tt}(\boldsymbol{\mathcal{A}}) = \mathbf{r} = (r_1, \ldots, r_{K-1})$ and has a left-orthogonal representation $\boldsymbol{\mathcal{A}} = [\boldsymbol{\mathcal{T}}_1, \ldots, \boldsymbol{\mathcal{T}}_K]$, then:*

$$\|\boldsymbol{\mathcal{A}}\|_* \leq \sqrt{r_1 \cdots r_{K-1}} \cdot \|\boldsymbol{\mathcal{A}}\|_{\mathrm{F}}.$$

Lemma B.2.4 (Theorem 1 of Tomioka and Suzuki (2014)). *For a random tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$ with mean-zero and independent sub-Gaussian entries with sub-Gaussian norm σ, its spectral norm satisfies:*

$$\|\boldsymbol{\mathcal{A}}\|_\sigma \leq \sqrt{8\sigma^2 \left[\bar{d} \log 5K + \log \frac{2}{\delta}\right]},$$

with probability at least $1 - \delta$.

B.3 Appendix for Section 3.5

B.3.1 Details of Simulation Setup

We summarize the data-generating model of all essential tensors involved in the simulation experiment in Table B.1.

B.3.2 Results on the Missing Propensity Estimation Error

We examine here the effectiveness of the RGrad algorithm for recovering the tensor parameter $\boldsymbol{\mathcal{B}}^*$ from a single observation $\boldsymbol{\mathcal{W}}$. We consider $d \in \{40, 60, 80, 100\}$ and $r \in \{3, 5, 7, 9\}$ when simulating $\boldsymbol{\mathcal{B}}^*$. For simulating $\boldsymbol{\mathcal{W}}$ using the Ising model, we fix $h(x) = x/2$ and consider either $g(x, y) \in \{0, xy/15\}$, where we term the case with $g = 0$ as the

Tensor	Generating Model	Additional Details
$\mathcal{B}^*$	$\mathcal{B}^* = \mathcal{C} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$ $\mathcal{C} \in \mathbb{R}^{r\times r\times r}, \mathbf{U}_i \in \mathbb{R}^{d_i \times r_i}$	$\mathcal{C} \overset{i.i.d.}{\sim} 0.5 \cdot \mathcal{N}(-1, 0.5) + 0.5 \cdot \mathcal{N}(1, 0.5)$ $\mathbf{U}_i = \begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 1 & \cdots & 1 \end{bmatrix}^\top$ and each row of $\mathbf{U}_i$ has $\lceil d_i/r_i \rceil$ ones.
$\mathcal{W}$	$p(\mathcal{W}) \propto \exp\left[-\mathcal{H}(\mathcal{W}\vert\mathcal{B}^*)\right]$ based on (3.7) and (3.8)	simulate by block-Gibbs MCMC, where in each proposal we first sample $\mathbb{I}_1 = \{(i_1, \ldots, i_K) \vert \sum_k i_k \text{ is odd}\}$ then $\mathbb{I}_1^c$. Each block is a Bernoulli model.
$\mathcal{X}$	$\mathcal{X} = \mathcal{X}^* + \mathcal{E}$	$\mathcal{X}$ is then masked by $\mathcal{W}$.
$\mathcal{X}^*$	$\mathcal{X}^* = \mathcal{C}^* \times_1 \mathbf{U}_1^* \times_2 \mathbf{U}_2^* \times_3 \mathbf{U}_3^*$ $\mathcal{C}^* \in \mathbb{R}^{3\times 3\times 3}, \mathbf{U}_i^* \in \mathbb{R}^{d_i \times 3}$	$\mathcal{C}^*, \mathbf{U}_1^*, \mathbf{U}_2^*, \mathbf{U}_3^* \overset{i.i.d.}{\sim} \mathcal{N}(0, 1)$
$\mathcal{E}$	$[\mathcal{E}]_s \overset{\text{independent}}{\sim} \mathcal{N}(0, \sigma_s^2)$	$\sigma_s = \begin{cases} 1 & \text{constant noise} \\ 0.5\left[1 + \exp(-\mathcal{B}_s^*)\right] & \text{adversarial noise} \end{cases}$

Table B.1: Details of the tensors generated in the simulation experiment.

(independent) Bernoulli model and the case with $g(x, y) = xy/15$ as the (product) Ising model. We split the training and calibration set randomly based on a $70\% - 30\%$ ratio.

Under each combination of the choices of (d, r, g), we generate $n = 30$ repetitions from a single chain of MCMC and fit RGrad to each repetition with the correctly specified $g(\cdot, \cdot)$ and a working rank $r' \in \{2, 3, \ldots, 15\}$. In Table B.2, we present the average rank selected by the P-AIC and P-BIC under the Bernoulli and Ising model with various (d, r) combinations.

Based on these numerical results, we find that the consistency of P-AIC and P-BIC depends on r/d, or the "low-rankness" $\mathcal{B}^*$. For tensors with high d and low r, both P-AIC and P-BIC are consistent, and the inconsistency emerges as r/d becomes larger. Generally speaking, P-AIC is more robust than P-BIC and is consistent across most of the simulation scenarios except for two cases with small tensor sizes. We therefore suggest using P-AIC for rank selection.

We then evaluate the fitted $\widehat{\mathcal{B}}$ with relative squared error (RSE) defined as: $\|\widehat{\mathcal{B}} - \mathcal{B}^*\|_{\mathrm{F}}/\|\mathcal{B}^*\|_{\mathrm{F}}$. The results, as plotted in Figure B.1, exhibit a tendency that as r/d becomes larger, so does the RSE, which echoes the results of the model selection. Additionally, the estimation error is lower for the Ising model, as compared to the Bernoulli model, given the same r and d. We interpret this result as that the Ising model estimator can leverage the additional information from neighbors to infer the missing propensity of each tensor

Bernoulli Model ($g(x,y)=0$)								
	P-AIC				P-BIC			
rank	$d=40$	$d=60$	$d=80$	$d=100$	$d=40$	$d=60$	$d=80$	$d=100$
$r=3$	**3.0**	**3.0**	**3.0**	**3.0**	2.0	2.0	**3.0**	**3.0**
$r=5$	**5.0**	**5.0**	**5.0**	**5.0**	2.0	2.1(0.3)	4.0	**5.0**
$r=7$	6.2(0.4)	**7.0**	**7.0**	**7.0**	2.0	2.0	2.0	2.3(0.4)
$r=9$	6.0(0.8)	**8.8(0.4)**	**9.0**	**9.0**	2.0	2.0	2.0	2.0
Ising Model ($g(x,y)=xy/15$)								
	P-AIC				P-BIC			
rank	$d=40$	$d=60$	$d=80$	$d=100$	$d=40$	$d=60$	$d=80$	$d=100$
$r=3$	**3.4(2.0)**	**3.0**	**3.0**	**3.0**	2.0	**3.0**	**3.0**	**3.0**
$r=5$	**7.7(4.1)**	**5.0**	**5.0**	**5.0**	2.0	4.0	**5.0**	**5.0**
$r=7$	13.9(0.2)	**7.0**	**7.0**	**7.0**	2.0	2.1(0.2)	4.0(0.2)	4.7(0.4)
$r=9$	13.9(0.2)	**9.0**	**9.0**	**9.0**	2.0	2.0	2.0	3.9(0.3)

Table B.2: Model selection result of the Bernoulli model and Ising model. Each number is the mean rank selected by P-AIC/P-BIC with $n = 30$ repetitions followed by its standard deviations, if non-zero. Boldface are the cases where the true rank is within 1.96 standard deviations of the average rank.

entry. In Figure 3.1(d) of the main paper, we plot the estimator for $\boldsymbol{\mathcal{B}}^*$ shown in 3.1(a) by RGrad based on a randomly chosen 70% training set and it is clear that $\widehat{\boldsymbol{\mathcal{B}}}$ mimics $\boldsymbol{\mathcal{B}}^*$ very well.

B.3.3 Results on Conformal Prediction Validation

As a companion result of Figure 3.2, we plot the empirical coverage and half of the average confidence interval width of three conformal prediction methods under different simulation scenarios in Figure B.2. The mis-coverage of the unweighted conformal prediction comes from under-coverage and is associated with shorter confidence intervals. The reason why unweighted conformal prediction has under-coverage is that under the adversarial noise setting, entries with higher missing propensity also have higher uncertainty, and using a uniform weight underestimates the uncertainty of a missing entry. As one can tell from Figure B.2, our CTC algorithm matches the oracle case quite well and provides well-calibrated confidence intervals.

Apart from these results, we also compared other binary tensor decomposition methods for estimating the missing propensity and conducting the weighted conformal prediction with our method. We mainly consider two competing methods other than the unweighted and oracle conformal prediction: 1) **GCP**: binary tensor decomposition with generalized CP-decomposition (Wang and Li, 2020; Hong et al., 2020); 2) **Tucker**: binary

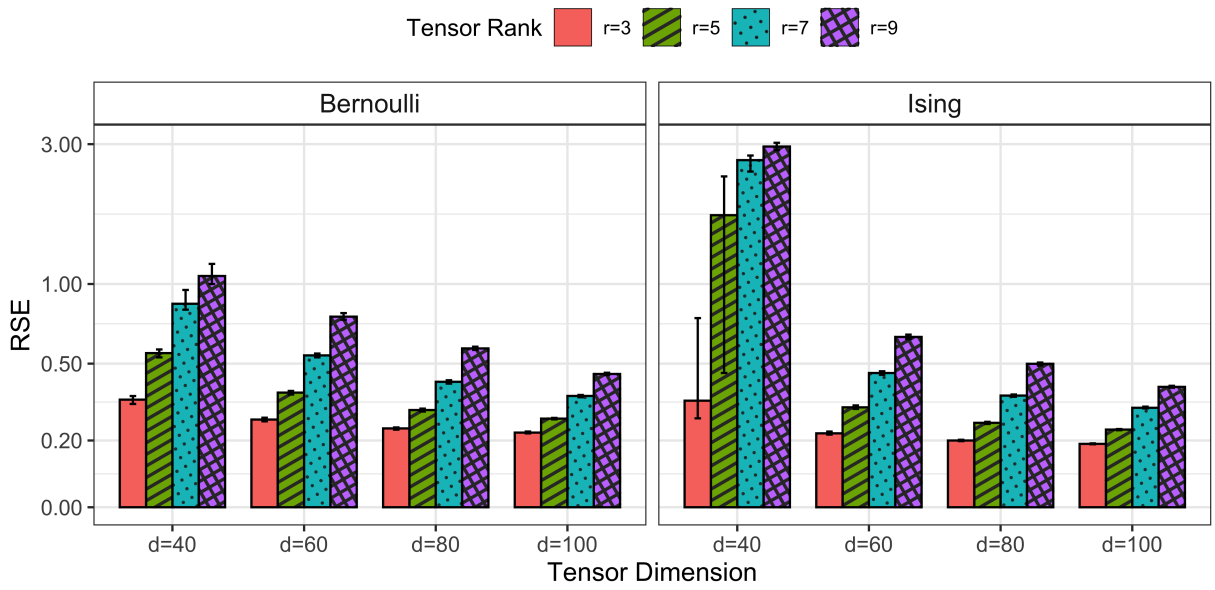


Figure B.1: Relative square error of the MPLE $\widehat{\boldsymbol{\mathcal{B}}}$ under the Bernoulli (left) and Ising model (right). The results are based on $n = 30$ repetitions with the working rank of each sample determined by P-AIC and each model is fitted by a randomly chosen 70% training set. Error bars show the 2.5% and 97.5% quantiles.

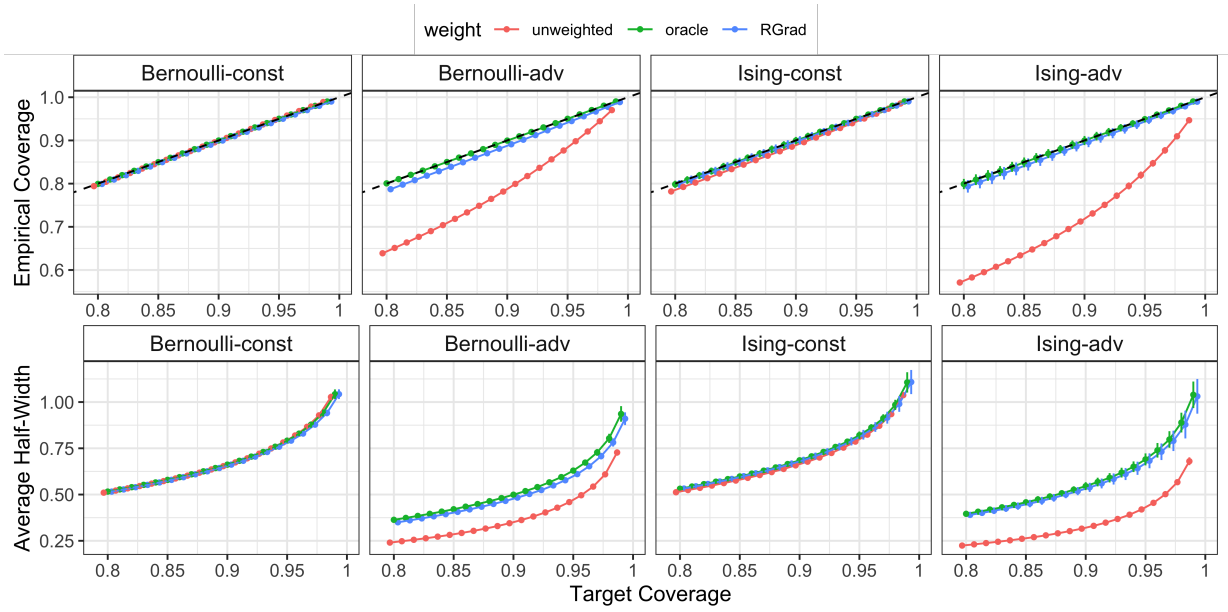


Figure B.2: Empirical coverage and average confidence interval half-width of the three conformal prediction methods across the Bernoulli and Ising model with constant (const) or adversarial (adv) noise. Results are based on $n = 30$ repetitions and error bars are ± 1.96 standard deviations.

tensor decomposition with generalized Tucker-decomposition (Lee and Wang, 2020; Cai et al., 2022c). Different from our approach, these two methods assume independence among all the binary entries and thus they are misspecified under the Ising model. We conduct **GCP** with gradient descent following (Hong et al., 2020) and **Tucker** with Riemannian gradient descent following (Cai et al., 2022c) and select the corresponding ranks using the BIC criterion, as suggested by the literature. We consider $r = 3, d \in \{40, 60\}$ and list the average mis-coverage % under the constant and adversarial noise regimes as well as the RSE of the estimated $\widehat{\boldsymbol{\mathcal{B}}}$ in Table B.3.

Our finding from Table B.3 is that our method consistently provides well-calibrated confidence intervals close to the oracle case and performs on average better than the GCP and Tucker method. Our mis-coverage % is statistically significantly better (p-value < 0.005) than the Tucker method under the adversarial noise regimes across different tensor dimensions and missingness generating models. The GCP method, surprisingly, provides confidence intervals close to our method but has significantly larger RSE for the estimator $\widehat{\boldsymbol{\mathcal{B}}}$. We found that CP-decomposition tends to underestimate the weights of the calibration data, therefore it has more testing data points with infinitely wide confidence intervals, making it less favorable.

(d, Model)	Method	constant noise mis-coverage %	adversarial noise mis-coverage %	RSE
(40, Bern)	unweighted	$0.463(0.244)$	$11.1(0.389)$	/
	oracle	$0.381(0.205)$	$0.409(0.232)$	/
	GCP	$0.373(0.183)$	$1.66(0.965)$	$0.522(0.069)$
	Tucker	$0.380(0.219)$	$0.841(0.431)$	$0.295(0.008)$
	RGrad	$0.377(0.235)$	$0.773(0.404)$	$0.345(0.010)$
(60, Bern)	unweighted	$0.401(0.165)$	$11.0(0.241)$	/
	oracle	$0.202(0.082)$	$0.207(0.105)$	/
	GCP	$0.203(0.092)$	$0.380(0.298)$	$0.281(0.036)$
	Tucker	$0.199(0.079)$	$0.842(0.231)$	$0.244(0.005)$
	RGrad	$0.200(0.078)$	$0.821(0.226)$	$0.271(0.004)$
(40, Ising)	unweighted	$1.19(0.298)$	$17.3(0.528)$	/
	oracle	$0.568(0.278)$	$0.666(0.331)$	/
	GCP	$0.870(0.597)$	$1.24(0.840)$	$1.81(0.621)$
	Tucker	$0.504(0.241)$	$1.80(0.653)$	$0.444(0.010)$
	RGrad	$0.713(0.377)$	$1.13(1.21)$	$\mathbf{0.341(0.304)}$
(60, Ising)	unweighted	$1.35(0.243)$	$17.2(0.310)$	/
	oracle	$0.302(0.136)$	$0.370(0.242)$	/
	GCP	$0.349(0.181)$	$0.638(0.506)$	$1.59(1.16)$
	Tucker	$0.329(0.216)$	$2.03(0.368)$	$0.404(0.007)$
	RGrad	$0.356(0.154)$	$0.580(0.339)$	$\mathbf{0.224(0.003)}$

Table B.3: Method comparisons of different conformal prediction methods with $r = 3$. The results include the average mis-coverage % defined in (3.25) under the constant (const.) and adversarial (adv.) noise regimes as well as the relatively squared error (RSE) of the estimator $\widehat{\boldsymbol{\mathcal{B}}}$.

APPENDIX C

Appendix for Chapter 4

This Appendix is organized as follows. In Section C.1, we prove Proposition 4.3.1 on the equivalence of the estimation problem of MARAC to a kernel ridge regression problem. In Section C.2, we prove Theorem 4.4.2 on the joint stationarity condition of the matrix and auxiliary vector time series. Then in Section C.3, we provide proofs of the theoretical results under fixed spatial dimensionality, including Proposition 4.4.4 and Theorem 4.4.5. In Section C.4, we present proofs of the theoretical results under high spatial dimensionality, namely Theorem 4.4.8. All essential lemmas used throughout the proofs are presented and proved in Section C.5. Finally, we include additional details of the simulation in Section C.6 as well as an approximated estimating algorithm for obtaining the penalized MLE via kernel truncation.

In this Appendix, we use $\bar{\rho}(\cdot)$, $\rho_i(\cdot)$, $\underline{\rho}(\cdot)$ and $\|\cdot\|_s$ to denote the maximum, i^{th} largest, minimum eigenvalue and spectral norm of a matrix. We use $a \vee b, a \wedge b$ to denote the maximum and minimum of a and b, respectively. For two sequences of random variables, say X_n, Y_n, we use $X_n \lesssim Y_n$ to denote the case where $X_n/Y_n = O_P(1)$, and $X_n \gtrsim Y_n$ to denote the case where $Y_n/X_n = O_P(1)$. We then use $X_n \asymp Y_n$ to denote the case where both $X_n \lesssim Y_n$ and $X_n \gtrsim Y_n$ hold.

C.1 Proof of Proposition 4.3.1

Proof. For each function $g_{q,d}(\cdot) \in \mathbb{H}_k$, we can decompose it as follows:

$$g_{q,d}(\cdot) = \sum_{s \in \mathbb{S}} \gamma_{q,d,s} k(\cdot, s) + \sum_{j=1}^{J} \alpha_{q,d,j} \phi_j(\cdot) + h_{q,d}(\cdot),$$

where $h_{q,d}(\cdot)$ does not belong to the null space of $\mathbb{H}_k$ nor the span of $\{k(\cdot, s) | s \in \mathbb{S}\}$. Here we assume that the null space of $\mathbb{H}_k$ contains only the zero function, so $\phi_j(\cdot) = 0,$ for all j.

By the reproducing property of the kernel $k(\cdot,\cdot)$, we have $\langle g_{q,d}, k(\cdot,s')\rangle_{\mathbb{H}_k} = g_{q,d}(s') = \sum_{s\in\mathbb{S}} \gamma_{q,d,s} k(s,s')$, which is independent of $h_{q,d}(\cdot)$, and therefore $h_{q,d}(\cdot)$ is independent of the prediction for $\mathbf{x}_t$ in the MARAC model. In addition, for any $h_{q,d}(\cdot) \notin \text{span}(\{k(\cdot,s)|s \in \mathbb{S}\})$, we have:

$$\|g_{q,d}\|^2_{\mathbb{H}_k} = \gamma_{q,d}^\top \mathbf{K}\gamma_{q,d} + \|h_{q,d}\|^2_{\mathbb{H}_k} \geq \|\sum_{s\in\mathbb{S}} \gamma_{q,d,s} k(\cdot,s)\|^2_{\mathbb{H}_k},$$

and the equality holds only if $h_{q,d}(\cdot) = 0$. Consequently, the global minimizer for the constrained optimization problem (4.7) must have $h_{q,d}(\cdot) = 0$. It then follows that the squared RKHS functional norm penalty for $g_{q,d}$ can be written as $\gamma_{q,d}^\top \mathbf{K}\gamma_{q,d}$ and the tensor coefficient $\boldsymbol{\mathcal{G}}_q$ satisfies $\text{vec}([\boldsymbol{\mathcal{G}}]_{::d}) = \mathbf{K}\gamma_{q,d}$. The remainder of the proof is straightforward by simple linear algebra and thus we omit it here. □

C.2 Proof of Theorem 4.4.2

Proof. Under Assumption 4.4.1 that the vector time series $\mathbf{z}_t$ follows a VAR($\widetilde{Q}$) process, we can derive that the vectorized matrix time series $\mathbf{X}_t$ and the vector time series $\mathbf{z}_t$ jointly follows a VAR($\max(P,Q,\widetilde{Q})$) process, namely,

$$\begin{bmatrix}\mathbf{x}_t\\ \mathbf{z}_t\end{bmatrix} = \sum_{l=1}^{\max(P,Q,\widetilde{Q})} \begin{bmatrix}(\mathbf{B}_l \otimes \mathbf{A}_l)\odot \mathbf{1}_{\{l\leq P\}} & \mathbf{G}_l^\top \odot \mathbf{1}_{\{l\leq Q\}}\\ \mathbf{O}_{D\times S} & \mathbf{C}_l \odot \mathbf{1}_{\{l\leq \widetilde{Q}\}}\end{bmatrix} \begin{bmatrix}\mathbf{x}_{t-l}\\ \mathbf{z}_{t-l}\end{bmatrix} + \begin{bmatrix}\mathbf{e}_t\\ \boldsymbol{\nu}_t\end{bmatrix}. \tag{C.1}$$

Let $L = \max(P,Q,\widetilde{Q})$ and $\mathbf{y}_t = [\mathbf{x}_t^\top, \mathbf{z}_t^\top]$. Denote the transition matrix in (C.1) at lag-l as $\mathbf{J}_l \in \mathbb{R}^{(S+D)\times(S+D)}$ and the error term as $\mathbf{u}_t^\top = [\mathbf{e}_t^\top, \boldsymbol{\nu}_t^\top]$, then we can rewrite the VAR(L) process in (C.1) as a VAR(1) process as:

$$\begin{bmatrix}\mathbf{y}_t\\ \mathbf{y}_{t-1}\\ \vdots\\ \mathbf{y}_{t-L+1}\end{bmatrix} = \begin{bmatrix}\mathbf{J}_1 & \mathbf{J}_2 & \cdots & \mathbf{J}_{L-1} & \mathbf{J}_L\\ \mathbf{I}_{S+D} & \mathbf{O}_{S+D} & \cdots & \cdots & \mathbf{O}_{S+D}\\ \mathbf{O}_{S+D} & \mathbf{I}_{S+D} & \mathbf{O}_{S+D} & \cdots & \mathbf{O}_{S+D}\\ \vdots & \vdots & \ddots & \ddots & \vdots\\ \mathbf{O}_{S+D} & \mathbf{O}_{S+D} & \cdots & \mathbf{I}_{S+D} & \mathbf{O}_{S+D}\end{bmatrix} \begin{bmatrix}\mathbf{y}_{t-1}\\ \mathbf{y}_{t-2}\\ \vdots\\ \mathbf{y}_{t-L}\end{bmatrix} + \begin{bmatrix}\mathbf{u}_t\\ \mathbf{0}_{S+D}\\ \vdots\\ \mathbf{0}_{S+D}\end{bmatrix}, \tag{C.2}$$

where we use $\mathbf{O}_{S+D}$ to denote a zero matrix of size $(S+D)\times(S+D)$. For this VAR(1) process to be stationary, we require that $\det(\lambda\mathbf{I} - \mathbf{J}) \neq 0$ for all $|\lambda| \geq 1, \lambda \in \mathbb{C}$, where $\mathbf{J}$ is the transition matrix in (C.2). The determinant $\det(\lambda\mathbf{I} - \mathbf{J})$ can be simplified by column

operations as:

$$
\begin{aligned}
&\det\left(\lambda\mathbf{I}-\mathbf{J}\right)\\
&=\det\begin{bmatrix}\lambda^{L}\mathbf{I}_{S}-\sum_{l=1}^{L}\lambda^{L-l}\left(\mathbf{B}_{l}\otimes\mathbf{A}_{l}\right)\odot\mathbf{1}_{\{l\leq P\}} & -\sum_{l=1}^{L}\lambda^{L-l}\mathbf{G}_{l}^{\top}\odot\mathbf{1}_{\{l\leq Q\}}\\ \mathbf{O} & \lambda^{L}\mathbf{I}_{D}-\sum_{l=1}^{L}\lambda^{L-l}\mathbf{C}_{l}\odot\mathbf{1}_{\{l\leq\widetilde{Q}\}}\end{bmatrix}\\
&=\lambda^{2L}\det\left[\mathbf{\Phi}_{1}(\lambda)\right]\det\left[\mathbf{\Phi}_{2}(\lambda)\right],
\end{aligned}
$$

where $\mathbf{\Phi}_1(\lambda)=\mathbf{I}_S-\sum_{p=1}^{P}\lambda^{-p}\left(\mathbf{B}_p\otimes\mathbf{A}_p\right)$ and $\mathbf{\Phi}_2(\lambda)=\mathbf{I}_D-\sum_{\widetilde{q}=1}^{\widetilde{Q}}\lambda^{-\widetilde{q}}\mathbf{C}_{\widetilde{q}}$, and setting $y=1/\lambda$ completes the proof. □

C.3 Theory under Fixed Spatial Dimension

C.3.1 Proof of Proposition 4.4.4

Proof. For the brevity of the presentation, we fix P, Q as 1 but the proofs presented below can be easily extended to an arbitrary P, Q. For the vectorized MARAC$(1,1)$ model (4.4), we can equivalently write it as:

$$\mathbf{x}_t=\mathbf{y}_t\boldsymbol{\theta}+\mathbf{e}_t, \tag{C.3}$$

where $\mathbf{y}_t=[\mathbf{x}_{t-1}^{\top}\otimes\mathbf{I}_S;\mathbf{z}_{t-1}^{\top}\otimes\mathbf{K}]$ and $\boldsymbol{\theta}=[\mathbf{vec}(\mathbf{B}_1\otimes\mathbf{A}_1)^{\top},\boldsymbol{\gamma}_1^{\top}]^{\top}$. Using $\mathbf{\Omega}=\mathbf{\Sigma}^{-1}$ to denote the precision matrix for $\mathbf{e}_t$, we can rewrite the penalized likelihood in (4.11) for $(\boldsymbol{\theta},\mathbf{\Omega})$ as:

$$h(\boldsymbol{\theta},\mathbf{\Omega})=-\frac{1}{2}\log|\mathbf{\Omega}|+\frac{1}{2}\mathrm{tr}\left(\mathbf{\Omega}\mathbf{S}(\boldsymbol{\theta})\right)+\frac{\lambda}{2}\boldsymbol{\theta}^{\top}\widetilde{\mathbf{K}}\boldsymbol{\theta}, \tag{C.4}$$

where $\mathbf{S}(\boldsymbol{\theta})=T^{-1}\sum_{t=1}^{T}(\mathbf{x}_t-\mathbf{y}_t\boldsymbol{\theta})(\mathbf{x}_t-\mathbf{y}_t\boldsymbol{\theta})^{\top}$, $\widetilde{\mathbf{K}}$ is defined as:

$$\widetilde{\mathbf{K}}=\begin{bmatrix}\mathbf{O}_{S\times S}\otimes\mathbf{K} & \mathbf{O}_{S\times D}\otimes\mathbf{K}\\ \mathbf{O}_{D\times S}\otimes\mathbf{K} & \mathbf{I}_D\otimes\mathbf{K}\end{bmatrix}.$$

We use $\boldsymbol{\theta}^*,\mathbf{\Omega}^*$ to denote the ground truth of $\boldsymbol{\theta},\mathbf{\Omega}$, respectively. We define $\mathbb{F}_{\boldsymbol{\theta}}$ and $\mathbb{F}_{\mathbf{\Omega}}$ as:

$$
\begin{aligned}
\mathbb{F}_{\boldsymbol{\theta}}&=\{[\mathbf{vec}(\mathbf{B}_1\otimes\mathbf{A}_1)^{\top},\boldsymbol{\gamma}_1^{\top}]^{\top}|\|\mathbf{A}_1\|_{\mathrm{F}}=1,\mathrm{sign}(\mathrm{tr}\left(\mathbf{A}_1\right))=1\};\\
\mathbb{F}_{\mathbf{\Omega}}&=\{\mathbf{\Sigma}_c^{-1}\otimes\mathbf{\Sigma}_r^{-1}|\mathbf{\Sigma}_r\in\mathbb{R}^{M\times M},\mathbf{\Sigma}_c\in\mathbb{R}^{N\times N},\underline{\rho}(\mathbf{\Sigma}_r),\underline{\rho}(\mathbf{\Sigma}_c)>0\}.
\end{aligned}
$$

The estimators of MARAC, denoted as $\widehat{\boldsymbol{\theta}},\widehat{\mathbf{\Omega}}$, is the minimizer of $h(\boldsymbol{\theta},\mathbf{\Omega})$ with $\boldsymbol{\theta}\in\mathbb{F}_{\boldsymbol{\theta}},\mathbf{\Omega}\in\mathbb{F}_{\mathbf{\Omega}}$.

In order to establish the consistency of $\widehat{\boldsymbol{\Sigma}} = \widehat{\boldsymbol{\Omega}}^{-1}$, it suffices to show that for any constant $c > 0$:

$$\mathrm{P}\left(\inf_{\|\bar{\boldsymbol{\Omega}}-\boldsymbol{\Omega}^*\|_{\mathrm{F}}\geq c}\inf_{\bar{\boldsymbol{\theta}}} h(\bar{\boldsymbol{\theta}},\bar{\boldsymbol{\Omega}}) \leq h(\boldsymbol{\theta}^*,\boldsymbol{\Omega}^*)\right) \to 0, \text{ as } T\to\infty. \tag{C.5}$$

This is because if (C.5) is established, then as $T \to \infty$ we have:

$$\mathrm{P}\left(\inf_{\|\bar{\boldsymbol{\Omega}}-\boldsymbol{\Omega}^*\|_{\mathrm{F}}\geq c}\inf_{\bar{\boldsymbol{\theta}}\in\mathbb{F}_{\boldsymbol{\theta}}} h(\bar{\boldsymbol{\theta}},\bar{\boldsymbol{\Omega}}) \geq \inf_{\|\bar{\boldsymbol{\Omega}}-\boldsymbol{\Omega}^*\|_{\mathrm{F}}\geq c}\inf_{\bar{\boldsymbol{\theta}}} h(\bar{\boldsymbol{\theta}},\bar{\boldsymbol{\Omega}}) > h(\boldsymbol{\theta}^*,\boldsymbol{\Omega}^*) \geq h(\widehat{\boldsymbol{\theta}},\widehat{\boldsymbol{\Omega}})\right)$$

approaching 1 and thus we must have $\|\widehat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}^*\|_{\mathrm{F}} < c$ with probability approaching 1 as $T \to \infty$, and the consistency is established since c is arbitrary.

To prove (C.5), we first fix $\boldsymbol{\Omega} = \bar{\boldsymbol{\Omega}}$ and let $\widetilde{\boldsymbol{\theta}}(\bar{\boldsymbol{\Omega}}) = \arg\min_{\boldsymbol{\theta}} h(\boldsymbol{\theta},\bar{\boldsymbol{\Omega}})$, thus we have:

$$\widetilde{\boldsymbol{\theta}}(\bar{\boldsymbol{\Omega}}) = \left(\frac{\sum_t \mathbf{y}_t^\top\bar{\boldsymbol{\Omega}}\mathbf{y}_t}{T} + \lambda\widetilde{\mathbf{K}}\right)^{-1}\left(\frac{\sum_t \mathbf{y}_t^\top\bar{\boldsymbol{\Omega}}\mathbf{x}_t}{T}\right), \tag{C.6}$$

which is a consistent estimator of $\boldsymbol{\theta}^*$ for any $\bar{\boldsymbol{\Omega}}$ given that $\lambda \to 0$ and the matrix and vector time series are covariance-stationary. To see that $\widetilde{\boldsymbol{\theta}}(\bar{\boldsymbol{\Omega}}) \xrightarrow{p} \boldsymbol{\theta}^*$, notice that:

$$\widetilde{\boldsymbol{\theta}}(\bar{\boldsymbol{\Omega}}) = (\mathbf{I} - \lambda\widetilde{\mathbf{K}})\boldsymbol{\theta}^* + \left(\frac{\sum_t \mathbf{y}_t^\top\bar{\boldsymbol{\Omega}}\mathbf{y}_t}{T} + \lambda\widetilde{\mathbf{K}}\right)^{-1}\left(\frac{\sum_t \mathbf{y}_t^\top\bar{\boldsymbol{\Omega}}\mathbf{e}_t}{T}\right), \tag{C.7}$$

and the first term converges to $\boldsymbol{\theta}^*$ since $\lambda = o(1)$. In the second term of (C.7), we have:

$$\frac{\sum_t \mathbf{y}_t^\top\bar{\boldsymbol{\Omega}}\mathbf{y}_t}{T} + \lambda\widetilde{\mathbf{K}} \xrightarrow{p} \begin{bmatrix} \boldsymbol{\Sigma}^*_{\mathbf{x},\mathbf{x}}\otimes\bar{\boldsymbol{\Omega}} & \boldsymbol{\Sigma}^*_{\mathbf{x},\mathbf{z}}\otimes\bar{\boldsymbol{\Omega}}\mathbf{K} \\ \boldsymbol{\Sigma}^*_{\mathbf{z},\mathbf{x}}\otimes\mathbf{K}\bar{\boldsymbol{\Omega}} & \boldsymbol{\Sigma}^*_{\mathbf{z},\mathbf{z}}\otimes\mathbf{K}\bar{\boldsymbol{\Omega}}\mathbf{K}\end{bmatrix}, \tag{C.8}$$

where $\boldsymbol{\Sigma}^*_{\mathbf{x},\mathbf{x}} = \mathrm{Var}(\mathbf{x}_t)$, $\boldsymbol{\Sigma}^*_{\mathbf{x},\mathbf{z}} = \mathrm{Cov}(\mathbf{x}_t,\mathbf{z}_t)$ and $\boldsymbol{\Sigma}^*_{\mathbf{z},\mathbf{z}} = \mathrm{Var}(\mathbf{z}_t)$. The convergence in probability in (C.8) holds due to the joint stationarity of $\mathbf{x}_t$ and $\mathbf{z}_t$ and the assumption that $\lambda = o(1)$. We further note that the sequence $\{\mathbf{y}_t^\top\bar{\boldsymbol{\Omega}}\mathbf{e}_t\}_{t=1}^T$ is a martingale difference sequence (MDS), and we have $\sum_{t=1}^T \mathbf{y}_t^\top\bar{\boldsymbol{\Omega}}\mathbf{e}_t/T = O_P(T^{-1/2})$ by the central limit theorem (CLT) of MDS (see proposition 7.9 of Hamilton (2020) for the central limit theorem of martingale difference sequence). Combining this result together with (C.8), we conclude that the second term in (C.7) is $o_P(1)$ and thus $\widetilde{\boldsymbol{\theta}}(\bar{\Omega})$ is consistent for $\boldsymbol{\theta}^*$.

Plugging $\widetilde{\boldsymbol{\theta}}(\bar{\boldsymbol{\Omega}})$ into $h(\boldsymbol{\theta},\bar{\boldsymbol{\Omega}})$ yields the profile likelihood of $\bar{\boldsymbol{\Omega}}$:

$$\ell(\bar{\boldsymbol{\Omega}}) = -\frac{1}{2}\log|\bar{\boldsymbol{\Omega}}| + \frac{1}{2}\mathrm{tr}\left(\bar{\boldsymbol{\Omega}}\frac{\sum_t \mathbf{x}_t[\mathbf{x}_t - \mathbf{y}_t\widetilde{\boldsymbol{\theta}}(\bar{\boldsymbol{\Omega}})]^\top}{T}\right).$$

To prove (C.5), it suffices to show that:

$$\mathrm{P}\left(\inf_{\|\bar{\mathbf{\Omega}}-\mathbf{\Omega}^*\|_{\mathrm{F}}\geq c}\ell(\bar{\mathbf{\Omega}})\leq\ell(\mathbf{\Omega}^*)\right)\to 0,\ \text{as } T\to\infty, \tag{C.9}$$

since $\ell(\mathbf{\Omega}^*)\leq h(\boldsymbol{\theta}^*,\mathbf{\Omega}^*)$. Now, since $\widetilde{\boldsymbol{\theta}}(\bar{\mathbf{\Omega}})\xrightarrow{p}\boldsymbol{\theta}^*$, we can write $\widetilde{\boldsymbol{\theta}}(\bar{\mathbf{\Omega}})=\boldsymbol{\theta}^*+\boldsymbol{\zeta}$, with $\|\boldsymbol{\zeta}\|_{\mathrm{F}}=o_P(1)$. Using this new notation, we can rewrite $\ell(\bar{\mathbf{\Omega}})$ as:

$$\begin{aligned}\ell(\bar{\mathbf{\Omega}}) &= -\frac{1}{2}\log|\bar{\mathbf{\Omega}}|+\frac{1}{2}\mathrm{tr}\left(\bar{\mathbf{\Omega}}\frac{\sum_t\mathbf{x}_t\mathbf{e}_t^\top}{T}\right)-\frac{1}{2}\mathrm{tr}\left(\left(\frac{\sum_t\mathbf{x}_t^\top\bar{\mathbf{\Omega}}\mathbf{y}_t}{T}\right)\boldsymbol{\zeta}\right) \\ &= \widetilde{\ell}(\bar{\mathbf{\Omega}})-\frac{1}{2}\mathrm{tr}\left(\left(\frac{\sum_t\mathbf{x}_t^\top\bar{\mathbf{\Omega}}\mathbf{y}_t}{T}\right)\boldsymbol{\zeta}\right),\end{aligned} \tag{C.10}$$

where we define the first two terms in (C.10) as $\widetilde{\ell}(\bar{\mathbf{\Omega}})$.

By the Cauchy-Schwartz inequality, we have:

$$\left|\frac{1}{2}\mathrm{tr}\left(\left(\frac{\sum_t\mathbf{x}_t^\top\bar{\mathbf{\Omega}}\mathbf{y}_t}{T}\right)\boldsymbol{\zeta}\right)\right|\leq\frac{1}{2}\left\|\frac{\sum_t\mathbf{x}_t^\top\bar{\mathbf{\Omega}}\mathbf{y}_t}{T}\right\|_{\mathrm{F}}\cdot\|\boldsymbol{\zeta}\|_{\mathrm{F}}. \tag{C.11}$$

By the definition of $\mathbf{y}_t$, we have:

$$\frac{\sum_t\mathbf{x}_t^\top\bar{\mathbf{\Omega}}\mathbf{y}_t}{T}=\left[\left(\frac{\sum_t\mathbf{x}_{t-1}\otimes\mathbf{x}_t}{T}\right)^\top(\mathbf{I}_S\otimes\bar{\mathbf{\Omega}});\left(\frac{\sum_t\mathbf{z}_{t-1}\otimes\mathbf{x}_t}{T}\right)^\top(\mathbf{I}_D\otimes\bar{\mathbf{\Omega}}\mathbf{K})\right],$$

and notice that $\mathbf{x}_{t-1}\otimes\mathbf{x}_t$ and $\mathbf{z}_{t-1}\otimes\mathbf{x}_t$ are just rearranged versions of $\mathbf{x}_t\mathbf{x}_{t-1}^\top$ and $\mathbf{x}_t\mathbf{z}_{t-1}^\top$, respectively. Therefore, by the joint stationarity of $\mathbf{x}_t$ and $\mathbf{z}_t$, we have the time average of $\mathbf{x}_{t-1}\otimes\mathbf{x}_t$ and $\mathbf{z}_{t-1}\otimes\mathbf{x}_t$ converging to the rearranged version of some constant auto-covariance matrices and therefore we have the term on the right-hand side of (C.11) being $o_P(1)$.

Given this argument, proving (C.9) is now equivalent to proving:

$$\mathrm{P}\left(\inf_{\|\bar{\mathbf{\Omega}}-\mathbf{\Omega}^*\|_{\mathrm{F}}\geq c}\widetilde{\ell}(\bar{\mathbf{\Omega}})\leq\widetilde{\ell}(\mathbf{\Omega}^*)\right)\to 0,\ \text{as } T\to\infty. \tag{C.12}$$

Define $\widetilde{\mathbf{\Omega}}$ as the unconstrained minimizer of $\widetilde{\ell}(\mathbf{\Omega})$, then explicitly, we have:

$$\begin{aligned}\widetilde{\mathbf{\Omega}}=\arg\min_{\mathbf{\Omega}}\widetilde{\ell}(\mathbf{\Omega}) &= \left(\frac{\sum_t\mathbf{e}_t\mathbf{x}_t^\top}{T}\right)^{-1} \\ &= \left(\frac{\sum_t\mathbf{e}_t\mathbf{e}_t^\top}{T}+\frac{\sum_t\mathbf{e}_t\left(\mathbf{y}_t\boldsymbol{\theta}^*\right)^\top}{T}\right)^{-1}\xrightarrow{p}\mathbf{\Omega}^*,\end{aligned}$$

where the final argument on the convergence in probability to $\mathbf{\Omega}^*$ is based on the fact that $\sum_{t=1}^{T} \mathbf{e}_t \left(\mathbf{y}_t \boldsymbol{\theta}^*\right)^\top / T = O_P(T^{-1/2})$ by the CLT of MDS. By the second-order Taylor expansion of $\widetilde{\ell}(\bar{\mathbf{\Omega}})$ at $\widetilde{\mathbf{\Omega}}$, we have:

$$\widetilde{\ell}(\bar{\mathbf{\Omega}}) = \widetilde{\ell}(\widetilde{\mathbf{\Omega}}) + \frac{1}{4} \mathbf{vec}(\bar{\mathbf{\Omega}} - \widetilde{\mathbf{\Omega}})^\top \left[\check{\mathbf{\Omega}}^{-1} \otimes \check{\mathbf{\Omega}}^{-1}\right] \mathbf{vec}(\bar{\mathbf{\Omega}} - \widetilde{\mathbf{\Omega}}), \tag{C.13}$$

where $\check{\mathbf{\Omega}} = \widetilde{\mathbf{\Omega}} + \eta(\bar{\mathbf{\Omega}} - \widetilde{\mathbf{\Omega}})$, for some $\eta \in [0, 1]$. For any constant $c > 0$ such that $\|\bar{\mathbf{\Omega}} - \mathbf{\Omega}^*\|_{\mathrm{F}} = c$, let $c = \kappa \bar{\rho}(\mathbf{\Omega}^*)$, where $\kappa > 0$ is also a constant that relates to c only. Consequently, we have:

$$\left|\bar{\rho}(\bar{\mathbf{\Omega}}) - \bar{\rho}(\mathbf{\Omega}^*)\right| \leq \|\bar{\mathbf{\Omega}} - \mathbf{\Omega}^*\|_s \leq \|\bar{\mathbf{\Omega}} - \mathbf{\Omega}^*\|_{\mathrm{F}} = \kappa \bar{\rho}(\mathbf{\Omega}^*),$$

and thus $\bar{\rho}(\bar{\mathbf{\Omega}}) \leq (1 + \kappa)\bar{\rho}(\mathbf{\Omega}^*)$. Conditioning on the event that $\|\bar{\mathbf{\Omega}} - \mathbf{\Omega}^*\|_{\mathrm{F}} = c$, we first have $\|\bar{\mathbf{\Omega}} - \widetilde{\mathbf{\Omega}}\|_{\mathrm{F}} \geq c/2$ to hold with probability approaching one, due to the consistency of $\widetilde{\mathbf{\Omega}}$. Furthermore, we also have:

$$\begin{aligned}
\underline{\rho}(\check{\mathbf{\Omega}}^{-1} \otimes \check{\mathbf{\Omega}}^{-1}) = \underline{\rho}(\check{\mathbf{\Omega}}^{-1})^2 &= \frac{1}{\bar{\rho}(\check{\mathbf{\Omega}})^2} \\
&\geq \left[\frac{1}{\bar{\rho}(\widetilde{\mathbf{\Omega}}) + \bar{\rho}(\bar{\mathbf{\Omega}})}\right]^2 \\
&\geq \left[\frac{1}{2\bar{\rho}(\mathbf{\Omega}^*) + (c + \bar{\rho}(\mathbf{\Omega}^*))}\right]^2 = \frac{1}{(3+\kappa)^2} \cdot \frac{1}{\bar{\rho}(\mathbf{\Omega}^*)^2},
\end{aligned}$$

where the last inequality holds with probability approaching one since $\mathrm{P}\left[\bar{\rho}(\widetilde{\mathbf{\Omega}}) \leq 2\bar{\rho}(\mathbf{\Omega}^*)\right] \to 1$. Utilizing these facts together with (C.13), we end up having:

$$\mathrm{P}\left[\widetilde{\ell}(\bar{\mathbf{\Omega}}) \geq \widetilde{\ell}(\widetilde{\mathbf{\Omega}}) + \frac{1}{16} \cdot \left(\frac{\kappa}{3+\kappa}\right)^2\right] \to 1, \text{ as } T \to \infty, \tag{C.14}$$

for any $\bar{\mathbf{\Omega}}$ such that $\|\bar{\mathbf{\Omega}} - \mathbf{\Omega}^*\|_{\mathrm{F}} = c = \kappa \bar{\rho}(\mathbf{\Omega}^*)$. Since κ is an arbitrary positive constant and $\widetilde{\ell}(\widetilde{\mathbf{\Omega}}) \xrightarrow{p} \widetilde{\ell}(\mathbf{\Omega}^*)$, we establish (C.12) and thereby completes the proof. □

C.3.2 Proof of Theorem 4.4.5

To prove Theorem 4.4.5, we first establish the consistency and the convergence rate of the estimators in Lemma C.3.1 below.

Lemma C.3.1. *Under the same assumption as Theorem 4.4.5, all model estimators for MARAC*

are $\sqrt{T}$-consistent, namely:

$$\|\widehat{\mathbf{A}}_p - \mathbf{A}_p^*\|_{\mathrm{F}} = O_P\left(\frac{1}{\sqrt{T}}\right), \|\widehat{\mathbf{B}}_p - \mathbf{B}_p^*\|_{\mathrm{F}} = O_P\left(\frac{1}{\sqrt{T}}\right), \|\widehat{\boldsymbol{\gamma}}_q - \boldsymbol{\gamma}_q^*\|_{\mathrm{F}} = O_P\left(\frac{1}{\sqrt{T}}\right),$$

for $p \in [P], q \in [Q]$. As a direct result, we also have:

$$\|\widehat{\mathbf{B}}_p \otimes \widehat{\mathbf{A}}_p - \mathbf{B}_p^* \otimes \mathbf{A}_p^*\|_{\mathrm{F}} = O_P\left(\frac{1}{\sqrt{T}}\right), \textit{ for } p \in [P].$$

We delay the proof of Lemma C.3.1 to Section C.5.2. With this lemma, we are now ready to present the proof of Theorem 4.4.5.

Proof. For the simplicity of notation and presentation, we fix P, Q as 1 but the proving technique can be generalized to arbitrary P, Q. To start with, we revisit the updating rule for $\mathbf{A}_p^{(l+1)}$ in (4.13). By plugging in the data-generating model for $\mathbf{X}_t$ according to MARAC$(1,1)$ model, we can transform (4.13) into:

$$\sum_{t\in[T]} \left[\Delta\mathbf{A}_1\mathbf{X}_{t-1}\widehat{\mathbf{B}}_1^\top + \mathbf{A}_1^*\mathbf{X}_{t-1}\Delta\mathbf{B}_1^\top + \Delta\boldsymbol{\mathcal{G}}_1 \bar{\times} \mathbf{z}_{t-1} - \mathbf{E}_t\right] \widehat{\boldsymbol{\Sigma}}_c^{-1}\widehat{\mathbf{B}}_1\mathbf{X}_{t-1}^\top = \mathbf{O}_{M\times M},$$

where for any arbitrary matrix/tensor $\mathbf{M}$, we define $\Delta\mathbf{M}$ as $\Delta\mathbf{M} = \widehat{\mathbf{M}} - \mathbf{M}^*$. One can simplify the estimating equation above by left multiplying $\widehat{\boldsymbol{\Sigma}}_r^{-1}$ and then vectorize both sides to obtain:

$$\begin{aligned}
&\sum_{t\in[T]} \left[(\mathbf{B}_1^*\mathbf{X}_{t-1}^\top)^\top(\boldsymbol{\Sigma}_c^*)^{-1}(\mathbf{B}_1^*\mathbf{X}_{t-1}^\top) \otimes (\boldsymbol{\Sigma}_r^*)^{-1}\right] \mathbf{vec}(\widehat{\mathbf{A}}_1 - \mathbf{A}_1^*) \\
+&\sum_{t\in[T]} \left[(\mathbf{B}_1^*\mathbf{X}_{t-1}^\top)^\top(\boldsymbol{\Sigma}_c^*)^{-1} \otimes (\boldsymbol{\Sigma}_r^*)^{-1}\mathbf{A}_1^*\mathbf{X}_{t-1}\right] \mathbf{vec}(\widehat{\mathbf{B}}_1^\top - (\mathbf{B}_1^*)^\top) \\
+&\sum_{t\in[T]} \left\{\mathbf{z}_{t-1}^\top \otimes \left[(\mathbf{B}_1^*\mathbf{X}_{t-1}^\top)^\top(\boldsymbol{\Sigma}_c^*)^{-1} \otimes (\boldsymbol{\Sigma}_r^*)^{-1}\mathbf{K}\right]\right\} \mathbf{vec}(\widehat{\boldsymbol{\gamma}}_1 - \boldsymbol{\gamma}_1^*) \\
=&\sum_{t\in[T]} \left[(\mathbf{B}_1^*\mathbf{X}_{t-1}^\top)^\top(\boldsymbol{\Sigma}_c^*)^{-1} \otimes (\boldsymbol{\Sigma}_r^*)^{-1}\right] \mathbf{vec}(\mathbf{E}_t) + o_P(\sqrt{T}).
\end{aligned}$$

On the left-hand side of the equation above, we replace $\widehat{\mathbf{B}}_1, \widehat{\boldsymbol{\Sigma}}_r, \widehat{\boldsymbol{\Sigma}}_c$ with their true values $\mathbf{B}_1^*, \boldsymbol{\Sigma}_r^*, \boldsymbol{\Sigma}_c^*$, since the discrepancies are of order $o_P(1)$ and can thus be incorporated into the

$o_P(\sqrt{T})$ term given the $\sqrt{T}$-consistency of $\widehat{\mathbf{A}}_1, \widehat{\mathbf{B}}_1, \widehat{\gamma}_1$. On the right-hand side, we have:

$$\begin{aligned}&\sum_t \mathbf{vec}(\widehat{\mathbf{\Sigma}}_r^{-1}\mathbf{E}_t\widehat{\mathbf{\Sigma}}_c^{-1}\widehat{\mathbf{B}}_1\mathbf{X}_{t-1}^\top)\\&= \sum_t \left[\mathbf{e}_t^\top \otimes (\mathbf{X}_{t-1}\otimes \mathbf{I}_M)\right] \mathbf{vec}\left[\left(\widehat{\mathbf{B}}_1^\top \otimes \mathbf{I}_M\right)\widehat{\mathbf{\Sigma}}^{-1}\right],\end{aligned}$$

where the process $\{\mathbf{e}_t^\top \otimes (\mathbf{X}_{t-1}\otimes \mathbf{I}_M)\}_{t=1}^T$ is a martingale difference sequence and the martingale central limit theorem (Hall and Heyde, 2014) implies that $\sum_t \left[\mathbf{e}_t^\top \otimes (\mathbf{X}_{t-1}\otimes \mathbf{I}_M)\right] = O_P(\sqrt{T})$, and thus by the consistency of $\widehat{\mathbf{\Sigma}}$ and $\widehat{\mathbf{B}}_1$, we can replace $\widehat{\mathbf{\Sigma}}$ and $\widehat{\mathbf{B}}_1$ with their true values and incorporate the remainders into $o_P(\sqrt{T})$.

Similar transformations can be applied to (4.14) and (4.15), where the penalty term is incorporated into $o_P(\sqrt{T})$ due to the assumption that $\lambda = o(T^{-\frac{1}{2}})$. With the notation that $\mathbf{U}_t = \mathbf{I}_N \otimes \mathbf{A}_1^*\mathbf{X}_{t-1}$, $\mathbf{V}_t = \mathbf{B}_1^*\mathbf{X}_{t-1}^\top \otimes \mathbf{I}_M$, $\mathbf{Y}_t = \mathbf{z}_{t-1}^\top \otimes \mathbf{K}$ and $\mathbf{W}_t = [\mathbf{V}_t; \mathbf{U}_t; \mathbf{Y}_t]$, these transformed estimating equations can be converted altogether into:

$$\begin{aligned}\left(\frac{1}{T}\sum_{t\in[T]}\mathbf{W}_t^\top(\mathbf{\Sigma}^*)^{-1}\mathbf{W}_t\right)\mathbf{vec}(\widehat{\mathbf{\Theta}}-\mathbf{\Theta}^*) = \frac{1}{T}\sum_{t\in[T]}\mathbf{W}_t^\top(\mathbf{\Sigma}^*)^{-1}\mathbf{vec}(\mathbf{E}_t)\\ + o_P(T^{-1/2}),\end{aligned} \tag{C.15}$$

where $\mathbf{vec}(\widehat{\mathbf{\Theta}}-\mathbf{\Theta}^*) = [\mathbf{vec}(\widehat{\boldsymbol{\mathcal{A}}}-\boldsymbol{\mathcal{A}}^*)^\top, \mathbf{vec}(\widehat{\boldsymbol{\mathcal{B}}}-\boldsymbol{\mathcal{B}}^*)^\top, \mathbf{vec}(\widehat{\boldsymbol{\mathcal{R}}}-\boldsymbol{\mathcal{R}}^*)^\top]^\top$, and $\widehat{\boldsymbol{\mathcal{A}}}, \widehat{\boldsymbol{\mathcal{B}}}, \widehat{\boldsymbol{\mathcal{R}}}$ are defined as $[\widehat{\boldsymbol{\mathcal{A}}}]_{::p} = \widehat{\mathbf{A}}_p$, $[\widehat{\boldsymbol{\mathcal{B}}}]_{::p} = \widehat{\mathbf{B}}_p^\top$, $[\widehat{\boldsymbol{\mathcal{R}}}]_{:dq} = \widehat{\gamma}_{q,d}$ and $\boldsymbol{\mathcal{A}}^*, \boldsymbol{\mathcal{B}}^*, \boldsymbol{\mathcal{R}}^*$ are the corresponding true coefficients.

In (C.15), we first establish that:

$$(1/T)\sum_{t\in[T]}\mathbf{W}_t^\top(\mathbf{\Sigma}^*)^{-1}\mathbf{W}_t \xrightarrow{p} \mathrm{E}\left[\mathbf{W}_t^\top(\mathbf{\Sigma}^*)^{-1}\mathbf{W}_t\right]. \tag{C.16}$$

To prove C.16, by the assumption that $\mathbf{X}_t$ and $\mathbf{z}_t$ are zero-meaned and jointly stationary, we have $T^{-1}\sum_{t\in[T]}\widetilde{\mathbf{x}}_t\widetilde{\mathbf{x}}_t^\top \xrightarrow{p} \mathrm{E}[\widetilde{\mathbf{x}}_t\widetilde{\mathbf{x}}_t^\top]$ by Lemma C.5.1 and Corollary C.5.2, where $\widetilde{\mathbf{x}}_t = [\mathbf{x}_t^\top, \mathbf{z}_t^\top]^\top$. See details of Lemma C.5.1 and Corollary C.5.2 in Section C.5.1. Then since each element of $\mathbf{W}_t^\top(\mathbf{\Sigma}^*)^{-1}\mathbf{W}_t$ is a linear combination of terms in $\widetilde{\mathbf{x}}_t\widetilde{\mathbf{x}}_t^\top$ (thus a continuous mapping), it is straightforward that (C.16) holds elementwise.

Given (C.16) and the fact that $\widehat{\mathbf{\Theta}}$ is $\sqrt{T}$-consistent, we can rewrite (C.15) as:

$$\begin{aligned}\mathrm{E}\left[\mathbf{W}_t^\top(\mathbf{\Sigma}^*)^{-1}\mathbf{W}_t\right]\mathbf{vec}(\widehat{\mathbf{\Theta}}-\mathbf{\Theta}^*) = \frac{1}{T}\sum_{t\in[T]}\mathbf{W}_t^\top(\mathbf{\Sigma}^*)^{-1}\mathbf{vec}(\mathbf{E}_t)\\ + o_P(T^{-1/2}),\end{aligned} \tag{C.17}$$

For the term on the right-hand side of (C.17), first notice that the sequence $\{\boldsymbol{\eta}_t\}_{t=1}^T$, where $\boldsymbol{\eta}_t = \mathbf{W}_t^\top(\boldsymbol{\Sigma}^*)^{-1}\mathbf{vec}(\mathbf{E}_t)$, is a zero-meaned, stationary vector martingale difference sequence (MDS), thanks to the independence of $\mathbf{E}_t$ from the jointly stationary $\mathbf{X}_{t-1}$ and $\mathbf{z}_{t-1}$. By the martingale central limit theorem (Hall and Heyde, 2014), we have:

$$\frac{1}{\sqrt{T}}\sum_{t\in[T]}\mathbf{W}_t^\top(\boldsymbol{\Sigma}^*)^{-1}\mathbf{vec}(\mathbf{E}_t) \xrightarrow{d.} \mathcal{N}(\mathbf{0}, \mathrm{E}\left[\mathbf{W}_t^\top(\boldsymbol{\Sigma}^*)^{-1}\mathbf{W}_t\right]). \tag{C.18}$$

Combining (C.17) and (C.18), we end up having:

$$\mathrm{E}\left[\mathbf{W}_t^\top(\boldsymbol{\Sigma}^*)^{-1}\mathbf{W}_t\right]\sqrt{T}\mathbf{vec}(\widehat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}^*) \xrightarrow{d.} \mathcal{N}(\mathbf{0}, \mathrm{E}\left[\mathbf{W}_t^\top(\boldsymbol{\Sigma}^*)^{-1}\mathbf{W}_t\right]). \tag{C.19}$$

The asymptotic distribution of $\sqrt{T}\mathbf{vec}(\widehat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}^*)$ can thus be derived by multiplying both sides of (C.19) by the inverse of $\mathbf{L} = \mathrm{E}\left[\mathbf{W}_t^\top(\boldsymbol{\Sigma}^*)^{-1}\mathbf{W}_t\right]$. However, the matrix $\mathbf{L}$ is not a full-rank matrix, because $\mathbf{L}\boldsymbol{\mu} = \mathbf{0}$, where $\boldsymbol{\mu} = [\mathbf{vec}(\boldsymbol{\mathcal{A}}^*)^\top, -\mathbf{vec}(\boldsymbol{\mathcal{B}}^*)^\top, \mathbf{0}^\top]^\top$. As a remedy, let $\boldsymbol{\zeta} = [\mathbf{vec}(\mathbf{A}_1^*)^\top\mathbf{0}^\top]^\top \in \mathbb{R}^{M^2+N^2+DMN}$, then given the identifiability constraint that $\|\mathbf{A}_1^*\|_{\mathrm{F}} = \|\widehat{\mathbf{A}}_1\|_{\mathrm{F}} = 1$ and the fact that $\widehat{\mathbf{A}}_1$ is $\sqrt{T}$-consistent, we have $\mathbf{vec}(\mathbf{A}_1^*)^\top\mathbf{vec}(\widehat{\mathbf{A}}_1 - \mathbf{A}_1^*) = o_P(T^{-1/2})$. Therefore, we have:

$$\sqrt{T}\boldsymbol{\zeta}^\top\mathbf{vec}(\widehat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}^*) \xrightarrow{p.} 0. \tag{C.20}$$

Combining (C.19) and (C.20) and using the Slutsky's theorem, we have $\mathbf{H}\sqrt{T}\mathbf{vec}(\widehat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}^*) \xrightarrow{d.} \mathcal{N}(\mathbf{0}, \mathbf{L})$, where $\mathbf{H} = \mathbf{L} + \boldsymbol{\zeta}\boldsymbol{\zeta}^\top$ and thus:

$$\sqrt{T}\mathbf{vec}(\widehat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}^*) \xrightarrow{d.} \mathcal{N}(\mathbf{0}, \mathbf{H}^{-1}\mathbf{L}\mathbf{H}^{-1}). \tag{C.21}$$

The final asymptotic distribution of $\mathrm{vec}(\widehat{\mathbf{B}}_1^\top) \otimes \mathrm{vec}(\widehat{\mathbf{A}}_1)$ and $\mathbf{K}\widehat{\boldsymbol{\gamma}}_{q,d}$ can be derived easily from (C.21) with multivariate delta method and we omit the details here. □

C.4 Theory under High Spatial Dimension

C.4.1 Proof of Theorem 4.4.8

Proof. In this proof, we will fix P, Q as 1 again for the ease of presentation but the technical details can be generalized to arbitrary P, Q. Since we fix the lags to be 1, we drop the subscript of the coefficients for convenience.

Under the specification of the MARAC$(1,1)$ model, we restate the model as:

$$\mathbf{x}_t = \left(\mathbf{x}_{t-1}^\top \otimes \mathbf{I}_S\right) \mathbf{vec}(\mathbf{B}^* \otimes \mathbf{A}^*) + \left(\mathbf{z}_{t-1}^\top \otimes \mathbf{K}\right) \boldsymbol{\gamma}^* + \mathbf{e}_t,$$

where $S = MN$ and we introduce the following additional notations:

$$\mathbf{Y}_T \coloneqq \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_T \end{bmatrix}, \quad \widetilde{\mathbf{X}}_T \coloneqq \begin{bmatrix} \mathbf{x}_0^\top \\ \vdots \\ \mathbf{x}_{T-1}^\top \end{bmatrix} \otimes \mathbf{I}_S, \quad \widetilde{\mathbf{z}}_T \coloneqq \begin{bmatrix} \mathbf{z}_0^\top \\ \vdots \\ \mathbf{z}_{T-1}^\top \end{bmatrix}, \quad \boldsymbol{\mathcal{E}}_T = \begin{bmatrix} \mathbf{e}_1 \\ \vdots \\ \mathbf{e}_T \end{bmatrix}.$$

We will drop the subscript T for convenience. Let $\boldsymbol{\phi}^* = \mathbf{vec}(\mathbf{B}^* \otimes \mathbf{A}^*)$, and $g_1^*, \ldots, g_D^* \in \mathbb{H}_k$ be the true autoregressive and functional parameters. Correspondingly, let $\boldsymbol{\gamma}_1^*, \ldots, \boldsymbol{\gamma}_D^*$ be the coefficients for the representers when evaluating $g_1^*, \ldots, g_D^*$ on a matrix grid, i.e. $\mathbf{K}\boldsymbol{\gamma}_d^*$ is a discrete evaluation of g_d^* on the matrix grid. Let $\mathbb{F}_{\boldsymbol{\phi}} = \{\mathbf{vec}(\mathbf{B} \otimes \mathbf{A}) | \|\mathbf{A}\|_{\mathrm{F}} = \mathrm{sign}(\mathrm{tr}\,(\mathbf{A})) = 1, \mathbf{A} \in \mathbb{R}^{M \times M}, \mathbf{B} \in \mathbb{R}^{N \times N}\}$. Using these new notations, the MARAC estimator is obtained by solving the following penalized least square problem:

$$\min_{\boldsymbol{\phi} \in \mathbb{F}_{\boldsymbol{\phi}}, \boldsymbol{\gamma} \in \mathbb{R}^{SD}} \mathfrak{L}_\lambda(\boldsymbol{\phi}, \boldsymbol{\gamma}) \coloneqq \left\{ \frac{1}{2T} \|\mathbf{Y} - \widetilde{\mathbf{X}}\boldsymbol{\phi} - (\widetilde{\mathbf{z}} \otimes \mathbf{K})\boldsymbol{\gamma}\|_{\mathrm{F}}^2 + \frac{\lambda}{2} \boldsymbol{\gamma}^\top (\mathbf{I}_D \otimes \mathbf{K}) \boldsymbol{\gamma} \right\}. \tag{C.22}$$

By fixing $\boldsymbol{\phi}$, the estimator for $\boldsymbol{\gamma}$ is given by $\widehat{\boldsymbol{\gamma}}(\boldsymbol{\phi}) = \arg\min_{\boldsymbol{\gamma}} \mathfrak{L}_\lambda(\boldsymbol{\phi}, \boldsymbol{\gamma})$, and can be explicitly written as:

$$\widehat{\boldsymbol{\gamma}}(\boldsymbol{\phi}) = T^{-1} \left[\widehat{\boldsymbol{\Sigma}}_{\mathbf{z}} \otimes \mathbf{K} + \lambda \cdot \mathbf{I}_{SD}\right]^{-1} \left(\widetilde{\mathbf{z}}^\top \otimes \mathbf{I}_S\right) \left(\mathbf{Y} - \widetilde{\mathbf{X}}\boldsymbol{\phi}\right). \tag{C.23}$$

Plugging (C.23) into (C.22) yields the profile likelihood for $\boldsymbol{\phi}$:

$$\ell_\lambda(\boldsymbol{\phi}) = \mathfrak{L}_\lambda(\boldsymbol{\phi}, \widehat{\boldsymbol{\gamma}}(\boldsymbol{\phi})) = \frac{1}{2T} \left(\mathbf{Y} - \widetilde{\mathbf{X}}\boldsymbol{\phi}\right)^\top \mathbf{W} \left(\mathbf{Y} - \widetilde{\mathbf{X}}\boldsymbol{\phi}\right), \tag{C.24}$$

where $\mathbf{W}$ is defined as:

$$\mathbf{W} = \left\{ \mathbf{I} - \frac{(\widetilde{\mathbf{z}} \otimes \mathbf{K}) \left[\widehat{\boldsymbol{\Sigma}}_{\mathbf{z}} \otimes \mathbf{K} + \lambda \cdot \mathbf{I}_{SD}\right]^{-1} \left(\widetilde{\mathbf{z}}^\top \otimes \mathbf{I}_S\right)}{T} \right\} = \left(\mathbf{I} + \frac{\widetilde{\mathbf{z}}\widetilde{\mathbf{z}}^\top}{\lambda T} \otimes \mathbf{K}\right)^{-1}, \tag{C.25}$$

and the second equality in (C.25) is by the Woodbury matrix identity. It can be seen that $\mathbf{W}$ is positive semi-definite and has all of its eigenvalues within $(0, 1)$. To improve the clarity and organization of the proof, we break down the proof into several major steps. In the first step, we establish the following result on $\widehat{\boldsymbol{\phi}}$:

Proposition C.4.1. *Under the assumptions of Theorem 4.4.8, we have:*

$$\left(\widehat{\boldsymbol{\phi}}-\boldsymbol{\phi}^*\right)^\top\left(\frac{\widetilde{\mathbf{X}}^\top\mathbf{W}\widetilde{\mathbf{X}}}{T}\right)\left(\widehat{\boldsymbol{\phi}}-\boldsymbol{\phi}^*\right)\lesssim O_P(C_g\lambda)+O_P(SD/T), \tag{C.26}$$

where $C_g=\sum_{d=1}^D\|g_d^*\|_{\mathbb{H}_k}^2$.

In order to derive the convergence rate of $\widehat{\boldsymbol{\phi}}$, we still require one additional result:

Lemma C.4.2. *Under the assumptions of Theorem 4.4.8 and the requirement that* $S\log S/T\to 0$, *it holds that:*

$$\underline{\rho}\left(\widetilde{\mathbf{X}}^\top\mathbf{W}\widetilde{\mathbf{X}}/T\right)\geq\frac{c_{0,S}}{2}>0, \tag{C.27}$$

with probability approaching 1 *as* $S,T\to\infty$, *where* $\underline{\rho}(\cdot)$ *is the minimum eigenvalue of a matrix and* $c_{0,S}=\underline{\rho}(\boldsymbol{\Sigma}^*_{\mathbf{x},\mathbf{x}}-\left(\boldsymbol{\Sigma}^*_{\mathbf{z},\mathbf{x}}\right)^\top\left(\boldsymbol{\Sigma}^*_{\mathbf{z},\mathbf{z}}\right)^{-1}\boldsymbol{\Sigma}^*_{\mathbf{z},\mathbf{x}})$.

The proof of Proposition C.4.1 and Lemma C.4.2 are relegated to Section C.4.2 and C.5.3, respectively. Combining Proposition C.4.1 and Lemma C.4.2, we can derive the error bound of $\widehat{\boldsymbol{\phi}}$ as:

$$\frac{1}{S}\|\widehat{\boldsymbol{\phi}}-\boldsymbol{\phi}^*\|_{\mathrm{F}}\lesssim O_P(\sqrt{\frac{C_g\gamma_S}{c_{0,S}S}})+O_P(\sqrt{\frac{D}{c_{0,S}TS}}). \tag{C.28}$$

Now with this error bound of the autoregressive parameter $\widehat{\boldsymbol{\phi}}$, we further derive the prediction error bound for the functional parameters. To start with, we have:

$$\begin{aligned}\frac{1}{\sqrt{TS}}\|(\widetilde{\mathbf{z}}\otimes\mathbf{K})(\widehat{\boldsymbol{\gamma}}-\boldsymbol{\gamma}^*)\|_{\mathrm{F}}&=\frac{1}{\sqrt{TS}}\left\|(\mathbf{I}-\mathbf{W})(\mathbf{Y}-\widetilde{\mathbf{X}}\widehat{\boldsymbol{\phi}})-(\widetilde{\mathbf{z}}\otimes\mathbf{K})\boldsymbol{\gamma}^*\right\|_{\mathrm{F}}\\&\leq\frac{1}{\sqrt{TS}}\Bigg[\underbrace{\|(\mathbf{I}-\mathbf{W})\boldsymbol{\mathcal{E}}\|_{\mathrm{F}}}_{J_1}+\underbrace{\|(\mathbf{I}-\mathbf{W})\widetilde{\mathbf{X}}(\widehat{\boldsymbol{\phi}}-\boldsymbol{\phi}^*)\|_{\mathrm{F}}}_{J_2}\\&\quad+\underbrace{\|\mathbf{W}(\widetilde{\mathbf{z}}\otimes\mathbf{K})\boldsymbol{\gamma}^*\|_{\mathrm{F}}}_{J_3}\Bigg],\end{aligned}$$

and we will bound the terms J_1, J_2, J_3 separately.

To bound J_1, we first establish two lemmas.

Lemma C.4.3. *Given the definition of* $\mathbf{W}$ *in* (C.25) *and under the assumptions of Theorem 4.4.8, we have* $O_P(\gamma_S^{-1/2r_0})\leq tr\,(\mathbf{I}-\mathbf{W})\leq O_P(\sqrt{S}\gamma_S^{-1/2r_0})$, *where* $\gamma_S=\lambda/S$. *Furthermore, we have* $tr\,(\mathbf{W})\leq SD$.

Lemma C.4.4. *Given the definition of* $\mathbf{W}$ *in* (C.25) *and under the assumptions of Theorem 4.4.8, we have that:*

$$\boldsymbol{\mathcal{E}}^\top \mathbf{W} \boldsymbol{\mathcal{E}} / tr\left(\mathbf{W}\right) = O_P(1).$$

Furthermore, we have $\boldsymbol{\mathcal{E}}^\top \left(\mathbf{I}-\mathbf{W}\right)^2 \boldsymbol{\mathcal{E}} / tr\left(\left(\mathbf{I}-\mathbf{W}\right)^2\right) = O_P(1)$.

We leave the proof of Lemma C.4.3 and Lemma C.4.4 to Section C.5.4 and C.5.5. By Lemma C.4.4, we have:

$$J_1^2 \asymp \text{tr}\left(\left(\mathbf{I}-\mathbf{W}\right)^2\right) \lesssim \text{tr}\left(\mathbf{I}-\mathbf{W}\right).$$

And by Lemma C.4.3, we have $J_1 \leq O_P(S^{1/4}\gamma_S^{-1/4r_0})$.

For J_2, we have the following bound:

$$\begin{aligned} J_2 &= \|(\mathbf{I}-\mathbf{W})\mathbf{W}^{-1/2}\mathbf{W}^{1/2}\widetilde{\mathbf{X}}(\widehat{\boldsymbol{\phi}}-\boldsymbol{\phi}^*)\|_{\text{F}} && \text{(C.29)} \\ &\leq \|(\mathbf{I}-\mathbf{W})\mathbf{W}^{-1/2}\|_s \cdot \|\mathbf{W}^{1/2}\widetilde{\mathbf{X}}(\widehat{\boldsymbol{\phi}}-\boldsymbol{\phi}^*)\|_{\text{F}} \\ &\leq \|\mathbf{W}^{-1/2}\|_s \cdot \|\mathbf{W}^{1/2}\widetilde{\mathbf{X}}(\widehat{\boldsymbol{\phi}}-\boldsymbol{\phi}^*)\|_{\text{F}}. && \text{(C.30)} \end{aligned}$$

To bound $\|\mathbf{W}^{-1/2}\|_s$, we can take advantage of the simpler form of $\mathbf{W}$ using the Woodbury matrix identity in (C.25) and obtain:

$$\begin{aligned} \|\mathbf{W}^{-1/2}\|_s = \bar{\rho}(\mathbf{W}^{-1})^{\frac{1}{2}} &= \bar{\rho}\left(\mathbf{I}+(\lambda T)^{-1}\widetilde{\mathbf{z}}\widetilde{\mathbf{z}}^\top \otimes \mathbf{K}\right)^{\frac{1}{2}} \\ &\leq \left[1+\lambda^{-1}\bar{\rho}(\mathbf{K})\bar{\rho}(T^{-1}\widetilde{\mathbf{z}}\widetilde{\mathbf{z}}^\top)\right]^{\frac{1}{2}} \leq \left[1+\lambda^{-1}\bar{\rho}(\mathbf{K})\text{tr}\left(\widehat{\boldsymbol{\Sigma}}_{\mathbf{z}}\right)\right]^{\frac{1}{2}}. \end{aligned}$$

In Lemma C.5.1, which we state later in Section C.5.1, we have shown that for N-dimensional stationary vector autoregressive process, the covariance estimator is consistent in the spectral norm as long as $N\log N/T \to 0$. Therefore, since $\{\mathbf{z}_t\}_{t=1}^T$ follows a stationary VAR($\widetilde{Q}$) process and its dimensionality D is fixed, we have $\|\widehat{\boldsymbol{\Sigma}}_{\mathbf{z}} - \boldsymbol{\Sigma}_{\mathbf{z}}^*\|_s \xrightarrow{p.} 0$ and thus with probability approaching 1, we have $\text{tr}(\widehat{\boldsymbol{\Sigma}}_{\mathbf{z}}) \leq 2\text{tr}(\boldsymbol{\Sigma}_{\mathbf{z}}^*)$. Therefore, we have $\|\mathbf{W}^{-1/2}\|_s \leq O_P(\sqrt{1+c_0/\lambda})$, where c_0 is a constant related to $\text{tr}\left(\boldsymbol{\Sigma}_{\mathbf{z}}^*\right)$ and $\bar{\rho}(\mathbf{K})$. Combining this with the result in Proposition C.4.1, we can bound J_2 via its upper bound (C.30) as:

$$J_2 \leq O_P\left(\sqrt{C_g\lambda T}\right) + O_P\left(\sqrt{C_g T}\right) + O_P(\sqrt{S}) + O_P\left(\sqrt{\gamma_S^{-1}}\right). \tag{C.31}$$

Finally, for J_3, we first notice that:

$$J_3 = \|\mathbf{W}(\widetilde{\mathbf{z}}\otimes\mathbf{K})\boldsymbol{\gamma}^*\|_{\text{F}} \leq \|\mathbf{W}^{1/2}\|_s \cdot \|\mathbf{W}^{1/2}(\widetilde{\mathbf{z}}\otimes\mathbf{K})\boldsymbol{\gamma}^*\|_{\text{F}} \leq \|\mathbf{W}^{1/2}(\widetilde{\mathbf{z}}\otimes\mathbf{K})\boldsymbol{\gamma}^*\|_{\text{F}}.$$

The upper bound of J_3 above can be further bounded by:

$$\begin{aligned}\|\mathbf{W}^{1/2}(\widetilde{\mathbf{z}} \otimes \mathbf{K})\boldsymbol{\gamma}^*\|_{\mathrm{F}}^2 &= (\lambda T)[(\mathbf{I}_D \otimes \mathbf{K})\boldsymbol{\gamma}^*]^\top \left\{\mathbf{I}_{SD} - \left(\lambda^{-1}\widehat{\boldsymbol{\Sigma}}_{\mathbf{z}} \otimes \mathbf{K} + \mathbf{I}_{SD}\right)^{-1}\right\}\boldsymbol{\gamma}^* \\ &= (\lambda T)\left(\sum_{d=1}^{D}\|g_d^*\|_{\mathbb{H}_k}^2\right) \\ &\quad - (\lambda^2 T)\,(\boldsymbol{\gamma}^*)^\top\left[(\mathbf{I}_D \otimes \mathbf{K})\left(\widehat{\boldsymbol{\Sigma}}_{\mathbf{z}} \otimes \mathbf{K} + \lambda\mathbf{I}_{SD}\right)^{-1}\right]\boldsymbol{\gamma}^* \\ &\leq C_g \lambda T, \end{aligned} \tag{C.32}$$

where $C_g = \sum_{d=1}^{D}\|g_d^*\|_{\mathbb{H}_k}^2$ is the norm of all the underlying functional parameters. The last inequality of (C.32) follows from the fact that the quadratic form led by $\lambda^2 T$ is non-negative. To see why, first note that:

$$(\mathbf{I}_D \otimes \mathbf{K})\left(\widehat{\boldsymbol{\Sigma}}_{\mathbf{z}} \otimes \mathbf{K} + \lambda\mathbf{I}_{SD}\right)^{-1} = \left(\widehat{\boldsymbol{\Sigma}}_{\mathbf{z}} \otimes \mathbf{I}_S\right)^{-1} - \left[\widehat{\boldsymbol{\Sigma}}_{\mathbf{z}} \otimes \mathbf{I}_S + \lambda^{-1}\widehat{\boldsymbol{\Sigma}}_{\mathbf{z}}^2 \otimes \mathbf{K}\right]^{-1}.$$

Then, we have the following lemma:

Lemma C.4.5. *If* $\mathbf{A}, \mathbf{B}$ *are symmetric, positive definite real matrices and* $\mathbf{A} - \mathbf{B}$ *is positive semi-definite, then* $\mathbf{B}^{-1} - \mathbf{A}^{-1}$ *is also positive semi-definite.*

We leave the proof to Section C.5.6. Let $\mathbf{M} = \widehat{\boldsymbol{\Sigma}}_{\mathbf{z}} \otimes \mathbf{I}_S + \lambda^{-1}\widehat{\boldsymbol{\Sigma}}_{\mathbf{z}}^2 \otimes \mathbf{K}$ and $\mathbf{N} = \widehat{\boldsymbol{\Sigma}}_{\mathbf{z}} \otimes \mathbf{I}_S$, then both $\mathbf{M}$ and $\mathbf{N}$ are positive definite and $\mathbf{M} - \mathbf{N}$ is positive semi-definite. By Lemma C.4.5, we have $\mathbf{N}^{-1} - \mathbf{M}^{-1}$ being positive semi-definite and thus (C.32) holds.

Using the result in (C.32), we eventually have $J_3 \leq O_P(\sqrt{C_g \lambda T})$. Combining all the bounds for J_1, J_2, J_3, we end up with:

$$\begin{aligned}\frac{1}{\sqrt{TS}}\|(\widetilde{\mathbf{z}} \otimes \mathbf{K})(\widehat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}^*)\|_{\mathrm{F}} \leq O_P\left(\frac{\sqrt{\gamma_S^{-1/2r_0}}}{\sqrt{T}\sqrt[4]{S}}\right) + O_P(\sqrt{\gamma_S}) \\ + O_P\left(\frac{1}{\sqrt{S}}\right) + O_P\left(\frac{\sqrt{\gamma_S^{-1}}}{\sqrt{TS}}\right),\end{aligned}$$

where we drop the term $O_P(T^{-1/2})$ as it is a higher order term of $O_P(S^{-1/2})$ under the condition that $S \log S/T \to 0$. □

C.4.2 Proof of Proposition C.4.1

Proof. The MARAC estimator $\widehat{\boldsymbol{\phi}}$ is the minimizer of $\ell_\lambda(\boldsymbol{\phi})$, defined in (C.24), for all $\boldsymbol{\phi} \in \mathbb{F}_\phi$ and thus $\ell_\lambda(\widehat{\boldsymbol{\phi}}) \leq \ell_\lambda(\boldsymbol{\phi}^*)$. Equivalently, this means that:

$$\frac{1}{2}\left(\widehat{\boldsymbol{\phi}}-\boldsymbol{\phi}^*\right)^\top\left(\frac{\widetilde{\mathbf{X}}^\top\mathbf{W}\widetilde{\mathbf{X}}}{T}\right)\left(\widehat{\boldsymbol{\phi}}-\boldsymbol{\phi}^*\right) \leq \frac{1}{T}\left[\left(\widetilde{\mathbf{z}}\otimes\mathbf{K}\right)\boldsymbol{\gamma}^*+\boldsymbol{\mathcal{E}}\right]^\top\mathbf{W}\widetilde{\mathbf{X}}\left(\widehat{\boldsymbol{\phi}}-\boldsymbol{\phi}^*\right).$$

Let $\boldsymbol{\delta} = \mathbf{W}^{1/2}\widetilde{\mathbf{X}}(\widehat{\boldsymbol{\phi}}-\boldsymbol{\phi}^*)/\sqrt{T}$ and $\boldsymbol{\omega} = \mathbf{W}^{1/2}\left[\left(\widetilde{\mathbf{z}}\otimes\mathbf{K}\right)\boldsymbol{\gamma}^*+\boldsymbol{\mathcal{E}}\right]/\sqrt{T}$, then the inequality can be simply written as $\boldsymbol{\delta}^\top\boldsymbol{\delta} \leq 2\boldsymbol{\delta}^\top\boldsymbol{\omega}$, and we can upper bound our quantity of interest, namely $\boldsymbol{\delta}^\top\boldsymbol{\delta}$, as:

$$\boldsymbol{\delta}^\top\boldsymbol{\delta} \leq 2(\boldsymbol{\delta}-\boldsymbol{\omega})^\top(\boldsymbol{\delta}-\boldsymbol{\omega})+2\boldsymbol{\omega}^\top\boldsymbol{\omega} \leq 4\boldsymbol{\omega}^\top\boldsymbol{\omega}.$$

Therefore, the bound of $\|\boldsymbol{\delta}\|_{\mathrm{F}}^2$ can be obtained via the bound of $\|\boldsymbol{\omega}\|_{\mathrm{F}}^2$. We have the following upper bound for $\|\boldsymbol{\omega}\|_{\mathrm{F}}^2$:

$$\begin{aligned}\|\boldsymbol{\delta}\|_{\mathrm{F}}^2 \leq 4\|\boldsymbol{\omega}\|_{\mathrm{F}}^2 &= \frac{4}{T}\left[\left(\widetilde{\mathbf{z}}\otimes\mathbf{K}\right)\boldsymbol{\gamma}^*+\boldsymbol{\mathcal{E}}\right]^\top\mathbf{W}\left[\left(\widetilde{\mathbf{z}}\otimes\mathbf{K}\right)\boldsymbol{\gamma}^*+\boldsymbol{\mathcal{E}}\right] \\ &\leq \frac{8}{T}\left[\underbrace{\|\mathbf{W}^{1/2}\left(\widetilde{\mathbf{z}}\otimes\mathbf{K}\right)\boldsymbol{\gamma}^*\|_{\mathrm{F}}^2}_{I_1}+\underbrace{\|\mathbf{W}^{1/2}\boldsymbol{\mathcal{E}}\|_{\mathrm{F}}^2}_{I_2}\right], \end{aligned} \tag{C.33}$$

where the last inequality follows from the fact that $\mathbf{W}$ is positive semi-definite.

For I_1, it can be bounded by (C.32) and thus $I_1 \leq C_g\lambda T$. To bound I_2, we utilize Lemma C.4.4 and bound I_2 as $I_2 \asymp \mathrm{tr}\left(\mathbf{W}\right) \leq SD$. Combining the bounds for I_1 and I_2, we have:

$$\|\boldsymbol{\delta}\|_{\mathrm{F}}^2 = \left(\widehat{\boldsymbol{\phi}}-\boldsymbol{\phi}^*\right)^\top\left(\frac{\widetilde{\mathbf{X}}^\top\mathbf{W}\widetilde{\mathbf{X}}}{T}\right)\left(\widehat{\boldsymbol{\phi}}-\boldsymbol{\phi}^*\right) \lesssim O_P(C_g\lambda)+O_P(SD/T),$$

which completes the proof. □

C.5 Technical Lemmas & Proofs

In this section, we first introduce Lemma C.5.1 on the consistency of the covariance matrix estimator for any stationary vector autoregressive process and then Corollary C.5.2 on the consistency of the covariance estimator of our MARAC model, given the joint stationarity condition. Then we provide proof for Lemma C.3.1 used in Section C.3.2 when proving Theorem 4.4.5 on the asymptotic normality under fixed spatial dimension. Then we pro-

vide proofs for Lemma C.4.2, C.4.3, C.4.4 and C.4.5 used in Section C.4 when proving the error bounds with high spatial dimensionality.

C.5.1 Statement of Lemma C.5.1

In Lemma C.5.1, we restate the result of Proposition 6 and 7 of Li and Xiao (2021), which covers the general result of the consistency of the estimator for the lag-0 auto-covariance matrix of a stationary VAR(p) process.

Lemma C.5.1. *Let $\mathbf{x}_t \in \mathbb{R}^N$ be a zero-meaned stationary VAR(p) process: $\mathbf{x}_t = \sum_{l=1}^{p} \mathbf{\Phi}_p \mathbf{x}_{t-p} + \boldsymbol{\xi}_t$, where $\boldsymbol{\xi}_t$ have independent sub-Gaussian entries. Let $\widehat{\mathbf{\Sigma}} = (1/T)\sum_{t=1}^{T} \mathbf{x}_t \mathbf{x}_t^\top$ and $\mathbf{\Sigma} = \mathrm{E}[\widehat{\mathbf{\Sigma}}]$, then we have:*

$$\mathrm{E}\|\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma}\|_s \leq C\left(\sqrt{\frac{N\log N}{T}} + \frac{N\log N}{T}\right)\|\mathbf{\Sigma}\|_s, \tag{C.34}$$

where C is an absolute constant.

We refer our readers to Appendix C.3 of Li and Xiao (2021) for the proof. As a corollary of Lemma C.5.1, we have the following results:

Corollary C.5.2. *Assume that $\{\mathbf{z}_t\}_{t=1}^T$ is generated by a stationary VAR($\widetilde{Q}$) process: $\mathbf{z}_t = \sum_{\tilde{q}=1}^{\widetilde{Q}} \mathbf{C}_{\tilde{q}} \mathbf{z}_{t-\tilde{q}} + \boldsymbol{\nu}_t$, with $\boldsymbol{\nu}_t$ having independent sub-Gaussian entries, then with $\widehat{\mathbf{\Sigma}}_{\mathbf{z}} = (1/T)\sum_{t=1}^{T} \mathbf{z}_t \mathbf{z}_t^\top$ and $\mathbf{\Sigma}_{\mathbf{z}}^* = \mathrm{E}[\widehat{\mathbf{\Sigma}}_{\mathbf{z}}]$, we have:*

$$\mathrm{P}\left(\left\|\widehat{\mathbf{\Sigma}}_{\mathbf{z}} - \mathbf{\Sigma}_{\mathbf{z}}^*\right\|_s \geq \epsilon\right) \leq C\epsilon^{-1}\left(\sqrt{\frac{D}{T}} + \frac{D}{T}\right), \tag{C.35}$$

with C being an absolute constant and ϵ being a fixed positive real number, and thus $\left\|\widehat{\mathbf{\Sigma}}_{\mathbf{z}} - \mathbf{\Sigma}_{\mathbf{z}}^*\right\|_s \xrightarrow{p} 0$.

Let $\{\mathbf{X}_t\}_{t=1}^T$ be a zero-meaned matrix time series generated by the MARAC model with lag P, Q and $\{\mathbf{z}_t\}_{t=1}^T$ satisfies the assumption above and $\{\mathbf{X}_t, \mathbf{z}_t\}_{t=1}^T$ are jointly stationary in the sense of Theorem 4.4.2. Assume further that $\mathbf{E}_t$ has i.i.d. Gaussian entries with constant variance σ^2, then for $\mathbf{y}_t = [\mathbf{x}_t^\top, \mathbf{z}_t^\top]^\top$, $\widehat{\mathbf{\Sigma}}_0 = (1/T)\sum_{t=1}^{T} \mathbf{y}_t \mathbf{y}_t^\top$ and $\mathbf{\Sigma}_0^* = \mathrm{E}[\mathbf{y}_t \mathbf{y}_t^\top]$, we have:

$$\mathrm{E}\left\|\widehat{\mathbf{\Sigma}}_0 - \mathbf{\Sigma}_0^*\right\|_s \leq C\left(\sqrt{\frac{S\log S}{T}} + \frac{S\log S}{T}\right)\|\mathbf{\Sigma}_0^*\|_s, \tag{C.36}$$

where C is an absolute constant.

Proof. The proof of (C.35) is straightforward from Lemma C.5.1 together with Markov inequality. The proof of (C.36) also follows from Lemma C.5.1 since $\{\mathbf{y}_t\}_{t=1}^T$ follows

a stationary VAR($\max(P, Q, \widetilde{Q})$) process with i.i.d. sub-Gaussian noise (see (C.1)) and $\mathrm{E}[(1/T)\sum_{t=1}^{T}\mathbf{y}_t\mathbf{y}_t^\top] = \mathrm{E}[\mathbf{y}_t\mathbf{y}_t^\top]$ due to stationarity. □

Note that the convergence of the variance estimator in spectral norm also indicates that each element of the variance estimator converges in probability. Also, the assumption that $\mathbf{E}_t$ has i.i.d. Gaussian entries can be relaxed to $\mathbf{E}_t$ having independent sub-Gaussian entries.

C.5.2 Proof of Lemma C.3.1

Proof. Without loss of generality, we fix P, Q as 1 and use the same notation as (C.3) in Section C.3.1, so the MARAC model can be written as $\mathbf{x}_t = \mathbf{y}_t\boldsymbol{\theta}^* + \mathbf{e}_t$. Correspondingly, the penalized log-likelihood $h(\boldsymbol{\theta}, \boldsymbol{\Omega})$ is specified by (C.4) and given any $\bar{\boldsymbol{\Omega}}$, we have $\widetilde{\boldsymbol{\theta}}(\bar{\boldsymbol{\Omega}}) = \arg\min_{\boldsymbol{\theta}} h(\boldsymbol{\theta}, \bar{\boldsymbol{\Omega}})$ as specified by (C.6). Given the decomposition of $\widetilde{\boldsymbol{\theta}}(\bar{\boldsymbol{\Omega}})$ in (C.7), we have:

$$\widetilde{\boldsymbol{\theta}}(\bar{\boldsymbol{\Omega}}) - \boldsymbol{\theta}^* = -\lambda\widetilde{\mathbf{K}}\boldsymbol{\theta}^* + \left(\frac{\sum_t \mathbf{y}_t^\top\bar{\boldsymbol{\Omega}}\mathbf{y}_t}{T} + \lambda\widetilde{\mathbf{K}}\right)^{-1}\left(\frac{\sum_t \mathbf{y}_t^\top\bar{\boldsymbol{\Omega}}\mathbf{e}_t}{T}\right),$$

where $\|\lambda\widetilde{\mathbf{K}}\boldsymbol{\theta}^*\|_{\mathrm{F}} = o(T^{-1/2})$ since $\lambda = o(T^{-1/2})$ and the norm of the second term is $O_P(T^{-1/2})$. To show that the norm of the second term is $O_P(T^{-1/2})$, we first observe that:

$$\begin{aligned}
&\left\|\left(\frac{\sum_t \mathbf{y}_t^\top\bar{\boldsymbol{\Omega}}\mathbf{y}_t}{T} + \lambda\widetilde{\mathbf{K}}\right)^{-1}\left(\frac{\sum_t \mathbf{y}_t^\top\bar{\boldsymbol{\Omega}}\mathbf{e}_t}{T}\right)\right\|_{\mathrm{F}} \\
&\leq \left\|\underbrace{\left(\frac{\sum_t \mathbf{y}_t^\top\bar{\boldsymbol{\Omega}}\mathbf{y}_t}{T} + \lambda\widetilde{\mathbf{K}}\right)^{-1}}_{\mathbf{L}_T^{-1}}\right\|_{\mathrm{F}} \cdot \left\|\underbrace{\left(\frac{\sum_t \mathbf{y}_t^\top\bar{\boldsymbol{\Omega}}\mathbf{e}_t}{T}\right)}_{\mathbf{R}_T}\right\|_{\mathrm{F}}.
\end{aligned}$$

For the sequence of random matrices $\{\mathbf{L}_T\}_{T=1}^{\infty}$, we have:

$$\mathbf{L}_T = \frac{\sum_t \mathbf{y}_t^\top\bar{\boldsymbol{\Omega}}\mathbf{y}_t}{T} + \lambda\widetilde{\mathbf{K}} \xrightarrow{p.} \begin{bmatrix} \mathrm{Cov}(\mathbf{x}_t, \mathbf{x}_t)\otimes\bar{\boldsymbol{\Omega}} & \mathrm{Cov}(\mathbf{x}_t, \mathbf{z}_t)\otimes\bar{\boldsymbol{\Omega}}\mathbf{K} \\ \mathrm{Cov}(\mathbf{z}_t, \mathbf{x}_t)\otimes\mathbf{K}\bar{\boldsymbol{\Omega}} & \mathrm{Cov}(\mathbf{z}_t, \mathbf{z}_t)\otimes\mathbf{K}\bar{\boldsymbol{\Omega}}\mathbf{K} \end{bmatrix},$$

and we define the limiting matrix as $\mathbf{L}$. To show this, first note that the covariance estimator $\widehat{\mathrm{Var}}([\mathbf{x}_t^\top, \mathbf{z}_t^\top]^\top) = T^{-1}\sum_t[\mathbf{x}_t^\top, \mathbf{z}_t^\top]^\top[\mathbf{x}_t^\top, \mathbf{z}_t^\top]$ converges in probability to the true covariance $\mathrm{Var}([\mathbf{x}_t^\top, \mathbf{z}_t^\top]^\top)$, which we prove separately in Corollary C.5.2. Secondly, notice that $\lambda = o(T^{-1/2})$, thus we have $\lambda\widetilde{\mathbf{K}} \to \mathbf{O}$ and thus we have the convergence in probability of $\mathbf{L}_T$ to $\mathbf{L}$ holds.

Notice that the limiting matrix $\mathbf{L}$ is invertible because the matrix $\mathbf{L}'$, defined as:

$$\mathbf{L}' = \begin{bmatrix} \mathbf{I} \otimes \mathbf{K} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} \end{bmatrix} \mathbf{L} \begin{bmatrix} \mathbf{I} \otimes \mathbf{K} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} \end{bmatrix} = \mathrm{Var}([\mathbf{x}_t^\top, \mathbf{z}_t^\top]^\top) \otimes (\mathbf{K}\bar{\mathbf{\Omega}}\mathbf{K}),$$

is invertible. To see why, firstly note that $\mathrm{Var}([\mathbf{x}_t^\top, \mathbf{z}_t^\top]^\top)$ is invertible because we can express $[\mathbf{x}_t^\top, \mathbf{z}_t^\top]^\top$ as $\sum_{j=0}^{\infty} \mathbf{\Phi}_j [\mathbf{e}_t^\top, \boldsymbol{\nu}_t^\top]^\top$, where $\{\mathbf{\Phi}_j\}_{j=0}^{\infty}$ is a sequence of matrices whose elements are absolutely summable and $\mathbf{\Phi}_0 = \mathbf{I}$, therefore, we have $\underline{\rho}(\mathrm{Var}([\mathbf{x}_t^\top, \mathbf{z}_t^\top]^\top)) \geq \underline{\rho}(\mathrm{Var}([\mathbf{e}_t^\top, \boldsymbol{\nu}_t^\top]^\top)) > 0$. Secondly, by Assumption 4.4.3, we have $\underline{\rho}(\mathbf{K}) > 0$ and we also have $\underline{\rho}(\bar{\mathbf{\Omega}}) > 0$ by definition, therefore we have $\mathbf{K}\bar{\mathbf{\Omega}}\mathbf{K}$ to be positive definite. The invertibility of $\mathbf{L}$ and the fact that $\mathbf{L}_T \xrightarrow{p.} \mathbf{L}$ indicates that $\mathbf{L}_T^{-1} \xrightarrow{p.} \mathbf{L}^{-1}$, since matrix inversion is a continuous function of the input matrix and the convergence in probability carries over under continuous transformations. Eventually, this leads to the conclusion that $\|\mathbf{L}_T^{-1}\|_{\mathrm{F}} = O_P(1)$.

For the sequence of random matrices $\{\mathbf{R}_T\}_{T=1}^{\infty}$, we note that the sequence $\{\mathbf{y}_t^\top \bar{\mathbf{\Omega}} \mathbf{e}_t\}_{t=1}^{\infty}$ is a martingale difference sequence (MDS) such that $\|\mathbf{R}_T\|_{\mathrm{F}} = O_P(T^{-1/2})$ (see proposition 7.9 of Hamilton (2020) for the central limit theorem of martingale difference sequence). Combining the result of $\|\mathbf{L}_T\|_{\mathrm{F}}$ and $\|\mathbf{R}_T\|_{\mathrm{F}}$, we conclude that $\|\widetilde{\boldsymbol{\theta}}(\bar{\mathbf{\Omega}}) - \boldsymbol{\theta}^*\|_{\mathrm{F}} = O_P(T^{-1/2})$.

Fix $\mathbf{\Omega} = \bar{\mathbf{\Omega}}$, we can decompose $h(\boldsymbol{\theta}, \bar{\mathbf{\Omega}})$ via the second-order Taylor expansion as follows:

$$\begin{aligned} h(\boldsymbol{\theta}, \bar{\mathbf{\Omega}}) &= h(\widetilde{\boldsymbol{\theta}}(\bar{\mathbf{\Omega}}), \bar{\mathbf{\Omega}}) + \frac{1}{2}(\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}(\bar{\mathbf{\Omega}}))^\top \left(\frac{\sum_t \mathbf{y}_t^\top \bar{\mathbf{\Omega}} \mathbf{y}_t}{T} + \lambda \widetilde{\mathbf{K}} \right) (\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}(\bar{\mathbf{\Omega}})) \\ &\geq h(\widetilde{\boldsymbol{\theta}}(\bar{\mathbf{\Omega}}), \bar{\mathbf{\Omega}}) + \frac{1}{2}\underline{\rho}(\mathbf{L}_T) \|\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}(\bar{\mathbf{\Omega}})\|_{\mathrm{F}}^2, \end{aligned} \tag{C.37}$$

and recall that $\mathbf{L}_T = T^{-1} \sum_t \mathbf{y}_t^\top \bar{\mathbf{\Omega}} \mathbf{y}_t + \lambda \widetilde{\mathbf{K}}$. In the previous proof, we've shown that $\mathbf{L}_T \xrightarrow{p.} \mathbf{L}$, with $\mathbf{L}$ being a positive definite matrix. Therefore, with probability approaching 1, we have $\underline{\rho}(\mathbf{L}_T) \geq \underline{\rho}(\mathbf{L})/2 > 0$.

With the lower bound on $\underline{\rho}(\mathbf{L}_T)$, we can claim that for some constant $C_1 > 0$:

$$\begin{aligned} &\inf_{\bar{\mathbf{\Omega}} \in \mathbb{F}_{\mathbf{\Omega}} : \|\bar{\mathbf{\Omega}} - \mathbf{\Omega}^*\|_{\mathrm{F}} \leq C_1} h(\boldsymbol{\theta}, \bar{\mathbf{\Omega}}) \\ &\geq \inf_{\bar{\mathbf{\Omega}} \in \mathbb{F}_{\mathbf{\Omega}} : \|\bar{\mathbf{\Omega}} - \mathbf{\Omega}^*\|_{\mathrm{F}} \leq C_1} \left\{ h(\widetilde{\boldsymbol{\theta}}(\bar{\mathbf{\Omega}}), \bar{\mathbf{\Omega}}) + \frac{1}{4}\underline{\rho}(\mathbf{L}) \cdot \|\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}(\bar{\mathbf{\Omega}})\|_{\mathrm{F}}^2 \right\}, \end{aligned} \tag{C.38}$$

with probability approaching 1. Now consider $\boldsymbol{\theta}$ belongs to the set $\{\boldsymbol{\theta} \in \mathbb{F}_{\boldsymbol{\theta}} | \sqrt{T}\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_{\mathrm{F}} \geq c_T\}$, where $c_T \to \infty$ is an arbitrary sequence that diverges to infinity. Within this set, we have:

$$\|\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}(\bar{\mathbf{\Omega}})\|_{\mathrm{F}} \geq \frac{c_T}{\sqrt{T}} - \|\boldsymbol{\theta}^* - \widetilde{\boldsymbol{\theta}}(\bar{\mathbf{\Omega}})\|_{\mathrm{F}}, \tag{C.39}$$

thus $\|\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}(\bar{\boldsymbol{\Omega}})\|_{\mathrm{F}} \gtrsim O_P(c_T'/\sqrt{T})$ for some sequence $c_T' \to \infty$ since $\|\widetilde{\boldsymbol{\theta}}(\bar{\boldsymbol{\Omega}}) - \boldsymbol{\theta}^*\|_{\mathrm{F}} = O_P(T^{-1/2})$. By the Taylor expansion in (C.37), we can conclude that $h(\boldsymbol{\theta}^*, \bar{\boldsymbol{\Omega}}) = h(\widetilde{\boldsymbol{\theta}}(\bar{\boldsymbol{\Omega}}), \bar{\boldsymbol{\Omega}}) + O_P(T^{-1})$, also using that $\|\widetilde{\boldsymbol{\theta}}(\bar{\boldsymbol{\Omega}}) - \boldsymbol{\theta}^*\|_{\mathrm{F}} = O_P(T^{-1/2})$. Combining this result together with the order of $\|\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}(\bar{\boldsymbol{\Omega}})\|_{\mathrm{F}}$, we have the following hold according to (C.38):

$$\mathrm{P}\left(\inf_{\sqrt{T}\|\boldsymbol{\theta}-\boldsymbol{\theta}^*\|_{\mathrm{F}}\geq c_T}\inf_{\bar{\boldsymbol{\Omega}}\in\mathbb{F}_{\boldsymbol{\Omega}}:\|\bar{\boldsymbol{\Omega}}-\boldsymbol{\Omega}^*\|_{\mathrm{F}}\leq C_1} h(\boldsymbol{\theta},\bar{\boldsymbol{\Omega}}) > \inf_{\bar{\boldsymbol{\Omega}}\in\mathbb{F}_{\boldsymbol{\Omega}}:\|\bar{\boldsymbol{\Omega}}-\boldsymbol{\Omega}^*\|_{\mathrm{F}}\leq C_1} h(\boldsymbol{\theta}^*,\bar{\boldsymbol{\Omega}})\right) \to 1. \tag{C.40}$$

The result in (C.40) indicates that for any $\boldsymbol{\theta}$ that lies outside of the set $\{\boldsymbol{\theta} \in \mathbb{F}_{\boldsymbol{\theta}} | \sqrt{T}\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_{\mathrm{F}} < c_T\}$, the penalized log-likelihood is no smaller than a sub-optimal solution with probability approaching 1. Therefore, with probability approaching 1, one must have $\sqrt{T}\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_{\mathrm{F}} \leq c_T$. And since the choice of c_T is arbitrary, we can conclude that $\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_{\mathrm{F}} = O_P(T^{-1/2})$ and thus each block of $\widehat{\boldsymbol{\theta}}$, namely $\widehat{\mathbf{A}}_p, \widehat{\mathbf{B}}_p, \widehat{\gamma}_q$ converges to their ground truth value at the rate of $T^{-1/2}$.

The convergence rate of $\widehat{\mathbf{B}}_p \otimes \widehat{\mathbf{A}_p}$ can be derived from the following inequality:

$$\|\widehat{\mathbf{B}}_p \otimes \widehat{\mathbf{A}}_p - \mathbf{B}_p^* \otimes \mathbf{A}_p^*\|_{\mathrm{F}} \leq \|\widehat{\mathbf{B}}_p\|_{\mathrm{F}} \cdot \|\widehat{\mathbf{A}}_p - \mathbf{A}_p^*\|_{\mathrm{F}} + \|\widehat{\mathbf{B}}_p - \mathbf{B}_p^*\|_{\mathrm{F}} \cdot \|\mathbf{A}_p^*\|_{\mathrm{F}},$$

as well as the convergence rate of $\widehat{\mathbf{A}}_p$ and $\widehat{\mathbf{B}}_p$.

□

C.5.3 Proof of Lemma C.4.2

Proof. Based on the definition of $\mathbf{W}$ in equation (C.25), we have

$$\begin{aligned}\frac{\widetilde{\mathbf{X}}^\top \mathbf{W} \widetilde{\mathbf{X}}}{T} &= \widehat{\boldsymbol{\Sigma}}_{\mathbf{x},\mathbf{x}} \otimes \mathbf{I}_S - \left(\widehat{\boldsymbol{\Sigma}}_{\mathbf{z},\mathbf{x}}^\top \otimes \mathbf{K}\right)\left(\widehat{\boldsymbol{\Sigma}}_{\mathbf{z},\mathbf{z}} \otimes \mathbf{K} + \lambda \mathbf{I}_{SD}\right)^{-1}\left(\widehat{\boldsymbol{\Sigma}}_{\mathbf{z},\mathbf{x}} \otimes \mathbf{I}_S\right) \\ &= \left(\widehat{\boldsymbol{\Sigma}}_{\mathbf{x},\mathbf{x}} - \widehat{\boldsymbol{\Sigma}}_{\mathbf{z},\mathbf{x}}^\top \widehat{\boldsymbol{\Sigma}}_{\mathbf{z},\mathbf{z}}^{-1} \widehat{\boldsymbol{\Sigma}}_{\mathbf{z},\mathbf{x}}\right) \otimes \mathbf{I}_S \\ &+ \left(\widehat{\boldsymbol{\Sigma}}_{\mathbf{z},\mathbf{x}} \otimes \mathbf{I}_S\right)^\top \left[\widehat{\boldsymbol{\Sigma}}_{\mathbf{z},\mathbf{z}}^2 \otimes \lambda^{-1}\mathbf{K} + \widehat{\boldsymbol{\Sigma}}_{\mathbf{z},\mathbf{z}} \otimes \mathbf{I}_S\right]^{-1}\left(\widehat{\boldsymbol{\Sigma}}_{\mathbf{z},\mathbf{x}} \otimes \mathbf{I}_S\right), \end{aligned} \tag{C.41}$$

where the second term in (C.41) is positive semi-definite since both $\underline{\rho}(\widehat{\boldsymbol{\Sigma}}_{\mathbf{z},\mathbf{z}})$ and $\underline{\rho}(\mathbf{K})$ are non-negative and the whole term is symmetric. Therefore, by Weyl's inequality, one can lower bound $\underline{\rho}(\widetilde{\mathbf{X}}^\top \mathbf{W} \widetilde{\mathbf{X}}/T)$ by $\underline{\rho}(\widehat{\boldsymbol{\Sigma}}_{\mathbf{x},\mathbf{x}} - \widehat{\boldsymbol{\Sigma}}_{\mathbf{z},\mathbf{x}}^\top \widehat{\boldsymbol{\Sigma}}_{\mathbf{z},\mathbf{z}}^{-1} \widehat{\boldsymbol{\Sigma}}_{\mathbf{z},\mathbf{x}})$. For simplicity, we will use $\mathbf{A}, \mathbf{B}, \mathbf{C}$ to denote $\boldsymbol{\Sigma}_{\mathbf{x},\mathbf{x}}^*, \boldsymbol{\Sigma}_{\mathbf{z},\mathbf{x}}^*, (\boldsymbol{\Sigma}_{\mathbf{z},\mathbf{z}}^*)^{-1}$, and $\widehat{\mathbf{A}}, \widehat{\mathbf{B}}, \widehat{\mathbf{C}}$ to denote $\widehat{\boldsymbol{\Sigma}}_{\mathbf{x},\mathbf{x}}, \widehat{\boldsymbol{\Sigma}}_{\mathbf{z},\mathbf{x}}, \widehat{\boldsymbol{\Sigma}}_{\mathbf{z},\mathbf{z}}^{-1}$, respectively. We will use $\widehat{\boldsymbol{\Sigma}}$ and $\boldsymbol{\Sigma}^*$ to denote the estimated and true covariance matrix of $[\mathbf{x}_t^\top, \mathbf{z}_t^\top]^\top$. It is evident that $\|\mathbf{A}\|_s \leq \|\boldsymbol{\Sigma}^*\|_s$ and $\|\mathbf{B}\|_s \leq \|\boldsymbol{\Sigma}^*\|_s$, since both $\mathbf{A}$ and $\mathbf{B}$ are blocks of $\boldsymbol{\Sigma}^*$ and can thus be represented as $\mathbf{E}_1^\top \boldsymbol{\Sigma}^* \mathbf{E}_2$ with $\mathbf{E}_1, \mathbf{E}_2$ being two block matrices with unity spectral norm.

The rest of the proof focuses on showing that with $S\log S/T \to 0$, $\underline{\rho}(\widehat{\mathbf{\Sigma}}_{\mathbf{x},\mathbf{x}} - \widehat{\mathbf{\Sigma}}_{\mathbf{z},\mathbf{x}}^\top \widehat{\mathbf{\Sigma}}_{\mathbf{z},\mathbf{z}}^{-1} \widehat{\mathbf{\Sigma}}_{\mathbf{z},\mathbf{x}}) \xrightarrow{p} \underline{\rho}(\mathbf{\Sigma}_{\mathbf{x},\mathbf{x}}^* - \left(\mathbf{\Sigma}_{\mathbf{z},\mathbf{x}}^*\right)^\top \left(\mathbf{\Sigma}_{\mathbf{z},\mathbf{z}}^*\right)^{-1} \mathbf{\Sigma}_{\mathbf{z},\mathbf{x}}^*)$. For brevity, we omit the subscript s for the spectral norm notation and simply use $\|\cdot\|$ in this proof.

To start with, we have:

$$\begin{aligned}
&\|\widehat{\mathbf{A}} - \widehat{\mathbf{B}}^\top \widehat{\mathbf{C}} \widehat{\mathbf{B}} - (\mathbf{A} - \mathbf{B}^\top \mathbf{C} \mathbf{B})\| \\
&\leq \|\widehat{\mathbf{A}} - \mathbf{A}\| + \|\widehat{\mathbf{B}}^\top \widehat{\mathbf{C}} \widehat{\mathbf{B}} - \mathbf{B}^\top \widehat{\mathbf{C}} \mathbf{B}\| + \|\mathbf{B}^\top \widehat{\mathbf{C}} \mathbf{B} - \mathbf{B}^\top \mathbf{C} \mathbf{B}\| \\
&\leq \|\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma}\| + \|(\widehat{\mathbf{B}} - \mathbf{B})^\top \widehat{\mathbf{C}} \widehat{\mathbf{B}}\| + \|\mathbf{B}^\top \mathbf{C} (\widehat{\mathbf{B}} - \mathbf{B})\| + \|\mathbf{B}^\top (\widehat{\mathbf{C}} - \mathbf{C}) \widehat{\mathbf{B}}\| \\
&\leq \|\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma}\| + \|\widehat{\mathbf{B}} - \mathbf{B}\| \cdot \left(\|\widehat{\mathbf{C}}\| \cdot \|\widehat{\mathbf{B}}\| + \|\mathbf{C}\| \cdot \|\mathbf{B}\|\right) \\
&+ \|\mathbf{B}\| \cdot \|\widehat{\mathbf{B}}\| \cdot \|\widehat{\mathbf{C}} - \mathbf{C}\|. \qquad \text{(C.42)}
\end{aligned}$$

Based on Corollary C.5.2, under the condition that $S\log S/T \to 0$ and the conditions that $\mathbf{z}_t$ follows a stationary VAR$(\widetilde{Q})$ process and is jointly stationary with $\mathbf{x}_t$, we have $\|\widehat{\mathbf{C}} - \mathbf{C}\| \xrightarrow{p} 0$ and $\|\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma}^*\| \xrightarrow{p} 0$. Therefore, with probability approaching 1, we have $\|\widehat{\mathbf{C}}\| \leq 2\|\mathbf{C}\|$, $\|\widehat{\mathbf{B}} - \mathbf{B}\| \leq \|\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma}^*\| \leq 2\|\mathbf{\Sigma}^*\|$ and $\|\widehat{\mathbf{B}}\| \leq 3\|\mathbf{\Sigma}^*\|$.

Combining these results and the upper bound in (C.42), with probability approaching 1, we have:

$$\begin{aligned}
\|\widehat{\mathbf{A}} - \widehat{\mathbf{B}}^\top \widehat{\mathbf{C}} \widehat{\mathbf{B}} - (\mathbf{A} - \mathbf{B}^\top \mathbf{C} \mathbf{B})\| \leq (1 + 7\|\mathbf{C}\| \cdot \|\mathbf{\Sigma}^*\|) \cdot \|\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma}^*\| \\
+ 3\|\mathbf{\Sigma}^*\|^2 \cdot \|\widehat{\mathbf{C}} - \mathbf{C}\|. \qquad \text{(C.43)}
\end{aligned}$$

The upper bound in (C.43) can be arbitrarily small as $S, T \to \infty$ since $\|\widehat{\mathbf{C}} - \mathbf{C}\| \xrightarrow{p} 0$ and $\|\widehat{\mathbf{\Sigma}} - \mathbf{\Sigma}^*\| \xrightarrow{p} 0$.

Eventually, with probability approaching 1, we have:

$$\underline{\rho}(\widehat{\mathbf{\Sigma}}_{\mathbf{x},\mathbf{x}} - \widehat{\mathbf{\Sigma}}_{\mathbf{z},\mathbf{x}}^\top \widehat{\mathbf{\Sigma}}_{\mathbf{z},\mathbf{z}}^{-1} \widehat{\mathbf{\Sigma}}_{\mathbf{z},\mathbf{x}}) \geq \frac{1}{2} \underline{\rho}\left(\mathbf{\Sigma}_{\mathbf{x},\mathbf{x}}^* - \left(\mathbf{\Sigma}_{\mathbf{z},\mathbf{x}}^*\right)^\top \left(\mathbf{\Sigma}_{\mathbf{z},\mathbf{z}}^*\right)^{-1} \mathbf{\Sigma}_{\mathbf{z},\mathbf{x}}^*\right) = \frac{c_{0,S}}{2}. \qquad \text{(C.44)}$$

This completes the proof.

□

C.5.4 Proof of Lemma C.4.3

Proof. By the definition of $\mathbf{W}$ in (C.25), we have:

$$\begin{aligned}\operatorname{tr}(\mathbf{I}-\mathbf{W}) &= \operatorname{tr}\left[\left(\widehat{\mathbf{\Sigma}}_{\mathbf{z}}\otimes\mathbf{K}+\lambda\mathbf{I}_{SD}\right)^{-1}\left(\widehat{\mathbf{\Sigma}}_{\mathbf{z}}\otimes\mathbf{K}\right)\right] \\ &= \sum_{s=1}^{S}\sum_{d=1}^{D}\frac{\rho_d(\widehat{\mathbf{\Sigma}}_{\mathbf{z}})\rho_s(\mathbf{K})}{\lambda+\rho_d(\widehat{\mathbf{\Sigma}}_{\mathbf{z}})\rho_s(\mathbf{K})} \leq D\cdot\sum_{s=1}^{S}\frac{1}{1+\lambda\bar{\rho}(\widehat{\mathbf{\Sigma}}_{\mathbf{z}})^{-1}\rho_s(\mathbf{K})^{-1}}. \end{aligned} \tag{C.45}$$

Using Lemma C.5.1, we can bound $\bar{\rho}(\widehat{\mathbf{\Sigma}}_{\mathbf{z}})$ by $2\bar{\rho}(\mathbf{\Sigma}^*_{\mathbf{z}})$ with probability approaching 1 as $T\to\infty$. Conditioning on this high probability event and using the Assumption 4.4.7 that the kernel function is separable, the kernel Gram matrix $\mathbf{K}$ can be written as $\mathbf{K}_2\otimes\mathbf{K}_1$ and thus (C.45) can be bounded as:

$$D\cdot\sum_{s=1}^{S}\frac{1}{1+\lambda\bar{\rho}(\widehat{\mathbf{\Sigma}}_{\mathbf{z}})^{-1}\rho_s(\mathbf{K})^{-1}} \leq D\cdot\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{1}{1+c_{\mathbf{z}}\lambda\rho_i(\mathbf{K}_1)^{-1}\rho_j(\mathbf{K}_2)^{-1}}, \tag{C.46}$$

where $c_{\mathbf{z}}=1/2\bar{\rho}(\mathbf{\Sigma}^*_{\mathbf{z}})$. As $M,N\to\infty$, based on Assumption 4.4.6, we have $\rho_i(\mathbf{K}_1)\to Mi^{-r_0}$ and $\rho_j(\mathbf{K}_2)\to Nj^{-r_0}$. Consequently, we can find two constants $0<c_1<c_2$, with c_1 being sufficiently small and c_2 being sufficiently large, such that:

$$\begin{aligned}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{1}{1+c_2\lambda(ij)^{r_0}/S} &\leq \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{1}{1+c_{\mathbf{z}}\lambda\rho_i(\mathbf{K}_1)^{-1}\rho_j(\mathbf{K}_2)^{-1}} \\ &\leq \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{1}{1+c_1\lambda(ij)^{r_0}/S}, \end{aligned} \tag{C.47}$$

where we, with a little abuse of notations, incorporate $c_{\mathbf{z}}$ into c_1, c_2. To estimate the order of the lower and upper bound in (C.47), we first notice that for any constant $c>0$, one has:

$$\sum_{i=1}^{M\wedge N}\frac{1}{1+c\lambda i^{2r_0}/S} \leq \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{1}{1+c\lambda(ij)^{r_0}/S} \leq 2(M\vee N)\sum_{i=1}^{M\vee N}\frac{1}{1+c\lambda i^{2r_0}/S}. \tag{C.48}$$

To approximate the sum in (C.48), notice that:

$$\sum_{i=1}^{M\vee N}\frac{1}{1+c\lambda i^{2r_0}/S} = (S/c\lambda)^{1/2r_0}\cdot\sum_{i=1}^{M\vee N}\frac{1}{1+[\frac{i}{(S/c\lambda)^{1/2r_0}}]^{2r_0}}\cdot\frac{1}{(S/c\lambda)^{1/2r_0}},$$

and furthermore, we have:

$$\lim_{S\to\infty}\sum_{i=1}^{M\vee N}\frac{1}{1+[\frac{i}{(S/c\lambda)^{1/2r_0}}]^{2r_0}}\cdot\frac{1}{(S/c\lambda)^{1/2r_0}}=\int_0^C\frac{1}{1+x^{2r_0}}\mathrm{d}x<\infty,$$

where $C=\lim_{S\to\infty}c(M\vee N)^{2r_0}\cdot\gamma_S$. In the assumptions of Theorem 4.4.8, we assume that $M\vee N=O(\sqrt{S})$ and $\lim_{S\to\infty}\gamma_S\cdot S^{r_0}\to C_1$ where $0<C_1\leq\infty$. As a result, we have C being either a finite value or infinity, thus we have:

$$\lim_{S\to\infty}\sum_{i=1}^{M\vee N}\frac{1}{1+c\lambda i^{2r_0}/S}=\int_0^C\frac{1}{1+x^{2r_0}}\mathrm{d}x\cdot\lim_{S\to\infty}(S/c\lambda)^{1/2r_0}=O(\gamma_S^{-1/2r_0}). \tag{C.49}$$

Combining (C.45), (C.46), (C.47) and (C.49), we have $\mathrm{tr}\,(\mathbf{I}-\mathbf{W})\lesssim O_P((M\vee N)\gamma_S^{-1/2r_0})=O_P(\sqrt{S}\gamma_S^{-1/2r_0})$. To obtain the lower bound of $\mathrm{tr}\,(\mathbf{I}-\mathbf{W})$, we have:

$$\mathrm{tr}\,(\mathbf{I}-\mathbf{W})\geq D\cdot\sum_{s=1}^{S}\frac{1}{1+\lambda c'_{\mathbf{z}}\rho_s(\mathbf{K})^{-1}}\geq D\cdot\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{1}{1+c_3\lambda(ij)^{r_0}/S},$$

which holds with probability approaching 1 and $c'_{\mathbf{z}}=2/\underline{\rho}(\mathbf{\Sigma}^*_{\mathbf{z}})$ and the second inequality follows from (C.47). To further lower bound the double summation, we have:

$$\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{1}{1+c_3\lambda(ij)^{r_0}/S}\geq\sum_{i=1}^{M\wedge N}\frac{1}{1+c_3\lambda(ij)^{r_0}/S}.$$

This new lower bound can be approximated with the same method as (C.49) under the assumption that $M\wedge N=O(\sqrt{S})$. We can obtain the lower bound of $\mathrm{tr}\,(\mathbf{I}-\mathbf{W})$ as $O_P(\gamma_S^{-1/2r_0})$, which establishes the final result.

The upper bound of $\mathrm{tr}\,(\mathbf{W})$ is trivial since:

$$\mathrm{tr}\,(\mathbf{W})=\sum_{s=1}^{S}\sum_{d=1}^{D}\frac{\lambda}{\lambda+\rho_d(\widehat{\mathbf{\Sigma}}_{\mathbf{z}})\rho_s(\mathbf{K})}\leq SD.$$

□

C.5.5 Proof of Lemma C.4.4

Proof. Given any fixed $\mathbf{W}$ and $t > 0$, let $K = \sqrt{8/3}\sigma$ and $c > 0$ be some constant, then we have:

$$\mathrm{P}\left[\left|\boldsymbol{\mathcal{E}}^\top \mathbf{W}\boldsymbol{\mathcal{E}} - \sigma^2 \mathrm{tr}\left(\mathbf{W}\right)\right| > t \middle| \mathbf{W}\right] \leq 2\exp\left[-c\min\left(\frac{t^2}{K^4\|\mathbf{W}\|_{\mathrm{F}}^2}, \frac{t}{K^2\|\mathbf{W}\|_s}\right)\right], \tag{C.50}$$

which holds by the Hanson-Wright inequality (Rudelson and Vershynin, 2013). Letting $t = \sigma^2 \mathrm{tr}\left(\mathbf{W}\right)/2$, $c_1 = 9c/256$, and using the fact that $\|\mathbf{W}\|_s \leq 1$, we have:

$$2\exp\left[-c\min\left(\frac{t^2}{K^4\|\mathbf{W}\|_{\mathrm{F}}^2}, \frac{t}{K^2\|\mathbf{W}\|_s}\right)\right] \leq 2\exp\left[-c_1 \mathrm{tr}\left(\mathbf{W}\right)\right]. \tag{C.51}$$

We can lower bound the trace of $\mathbf{W}$ as follows. First, note that:

$$\mathrm{tr}\left(\mathbf{W}\right) = \sum_{s=1}^{S}\sum_{d=1}^{D} \frac{\lambda}{\lambda + \rho_d(\widehat{\boldsymbol{\Sigma}}_{\mathbf{z}})\rho_s(\mathbf{K})} \geq SD \cdot \frac{\lambda}{\lambda + \bar{\rho}(\widehat{\boldsymbol{\Sigma}}_{\mathbf{z}})\bar{\rho}(\mathbf{K})}.$$

By the assumption that $\bar{\rho}(\mathbf{K})$ is bounded and that the fact that $\bar{\rho}(\widehat{\boldsymbol{\Sigma}}_{\mathbf{z}}) \leq 2\bar{\rho}(\boldsymbol{\Sigma}_{\mathbf{z}}^*)$ with probability approaching 1 as $T \to \infty$, we have:

$$\mathrm{P}\left[\mathrm{tr}\left(\mathbf{W}\right) \geq \frac{SD\lambda}{\lambda + \bar{c}}\right] \to 1, \quad \text{as } T \to \infty, \tag{C.52}$$

where $\bar{c} = 2\bar{\rho}(\boldsymbol{\Sigma}_{\mathbf{z}}^*)\bar{\rho}(\mathbf{K})$. Since $r_0 < 2$ and $\gamma_S \cdot S^{r_0} \to C_1$ as $S \to \infty$, with C_1 being either a positive constant or infinity, we have $\gamma_S \cdot S^2 = \lambda \cdot S \to \infty$. Therefore, we have $\mathrm{tr}\left(\mathbf{W}\right) \to \infty$ with probability approaching 1, as $S, T \to \infty$.

With these results, we can now upper bound the unconditional probability of the event $\{\left|\boldsymbol{\mathcal{E}}^\top \mathbf{W}\boldsymbol{\mathcal{E}} - \sigma^2 \mathrm{tr}\left(\mathbf{W}\right)\right| > \sigma^2 \mathrm{tr}\left(\mathbf{W}\right)/2\}$ as follows:

$$\begin{aligned}
&\mathrm{P}\left[\left|\boldsymbol{\mathcal{E}}^\top \mathbf{W}\boldsymbol{\mathcal{E}} - \sigma^2 \mathrm{tr}\left(\mathbf{W}\right)\right| > \sigma^2 \mathrm{tr}\left(\mathbf{W}\right)/2\right] \\
&\leq \mathrm{E}\left[2\exp\left[-c\mathrm{tr}\left(\mathbf{W}\right)\right]\right] \\
&\leq 2\left\{1 \cdot \mathrm{P}\left(\mathrm{tr}\left(W\right) < \frac{SD\lambda}{\lambda + \bar{c}}\right) + \exp\left[-c\frac{SD\lambda}{\lambda + \bar{c}}\right] \cdot \mathrm{P}\left(\mathrm{tr}\left(W\right) \geq \frac{SD\lambda}{\lambda + \bar{c}}\right)\right\} \to 0.
\end{aligned} \tag{C.53}$$

This indicates that $\boldsymbol{\mathcal{E}}^\top \mathbf{W}\boldsymbol{\mathcal{E}} \asymp \mathrm{tr}\left(\mathbf{W}\right)$.

The proof of $\boldsymbol{\mathcal{E}}^\top \left(\mathbf{I} - \mathbf{W}\right)^2 \boldsymbol{\mathcal{E}} \asymp \mathrm{tr}\left(\left(\mathbf{I} - \mathbf{W}\right)^2\right)$ is similar to the proof above. In the first

step, similar to (C.50) and (C.51), we have the following tail probability bound:

$$\begin{aligned}&\mathrm{P}\left[\left|\boldsymbol{\mathcal{E}}^{\top}\left(\mathbf{I}-\mathbf{W}\right)^{2}\boldsymbol{\mathcal{E}}-\sigma^{2}\mathrm{tr}\left(\left(\mathbf{I}-\mathbf{W}\right)^{2}\right)\right|>\frac{\sigma^{2}\mathrm{tr}\left(\left(\mathbf{I}-\mathbf{W}\right)^{2}\right)}{2}\middle|\mathbf{W}\right]\\&\leq 2\exp\left\{-c\mathrm{tr}\left(\left(\mathbf{I}-\mathbf{W}\right)^{2}\right)\right\}.\end{aligned}\tag{C.54}$$

We can actually establish the unboundedness of $\mathrm{tr}\left(\left(\mathbf{I}-\mathbf{W}\right)^{2}\right)$ by following the same idea as the proof for Lemma C.4.3, where we have:

$$\mathrm{tr}\left(\left(\mathbf{I}-\mathbf{W}\right)^{2}\right)\geq\left(S/c\lambda\right)^{1/2r_0}\cdot\sum_{i=1}^{M\wedge N}\left\{\frac{1}{1+\left[\frac{i}{\left(S/c\lambda\right)^{1/2r_0}}\right]^{2r_0}}\right\}^{2}\left(S/c\lambda\right)^{-1/2r_0},$$

with probability approaching 1 and c is some constant. Therefore, we have $\mathrm{tr}\left(\left(\mathbf{I}-\mathbf{W}\right)^{2}\right)\gtrsim O_P(\gamma_S^{-1/2r_0})$. The rest of the proof follows the idea of (C.53) and we omit the details here. □

C.5.6 Proof of Lemma C.4.5

Proof. For any two arbitrary symmetric matrices $\mathbf{M},\mathbf{N}$ with identical sizes, we use $\mathbf{M}\gtrsim\mathbf{N}$ to indicate that $\mathbf{M}-\mathbf{N}$ is positive semi-definite and we use $\mathbf{M}^{1/2}$ to denote the symmetric, positive semi-definite square root matrix of $\mathbf{M}$.

Since $\mathbf{A}-\mathbf{B}$ is positive semi-definite, multiplying it by $\mathbf{B}^{-1/2}$ on both left and right sides of $\mathbf{A}-\mathbf{B}$, we have $\mathbf{B}^{-1/2}\mathbf{A}\mathbf{B}^{-1/2}\gtrsim\mathbf{I}$. Therefore, we have $\mathbf{B}^{-1/2}\mathbf{A}^{1/2}\mathbf{A}^{1/2}\mathbf{B}^{-1/2}\gtrsim\mathbf{I}$. Notice that the matrix $\mathbf{A}^{1/2}\mathbf{B}^{-1/2}$ is invertible and thus has no zero eigenvalues. As a result, all eigenvalues of $\mathbf{B}^{-1/2}\mathbf{A}^{1/2}\mathbf{A}^{1/2}\mathbf{B}^{-1/2}$ are the same as the eigenvalues of $\mathbf{A}^{1/2}\mathbf{B}^{-1/2}\mathbf{B}^{-1/2}\mathbf{A}^{1/2}$ and thus $\mathbf{A}^{1/2}\mathbf{B}^{-1/2}\mathbf{B}^{-1/2}\mathbf{A}^{1/2}\gtrsim\mathbf{I}$. Multiplying both sides by $\mathbf{A}^{-1/2}$ on both the left and right sides yields $\mathbf{B}^{-1}\gtrsim\mathbf{A}^{-1}$, which completes the proof. □

C.6 Additional Details on Simulation and Algorithm

C.6.1 Simulation Setup

We generate the simulated dataset according to the MARAC(P,Q) model specified by (4.1) and (4.3). We simulate the autoregressive coefficients $\mathbf{A}_p,\mathbf{B}_p$ such that they satisfy the stationarity condition specified in Theorem 4.4.2 and have a banded structure. We use a similar setup for generating $\boldsymbol{\Sigma}_r,\boldsymbol{\Sigma}_c$ with their diagonals fixed at unity. In Figure C.1, we plot the simulated $\mathbf{A}_1,\mathbf{B}_1,\boldsymbol{\Sigma}_r,\boldsymbol{\Sigma}_c$ when $(M,N)=(20,20)$.

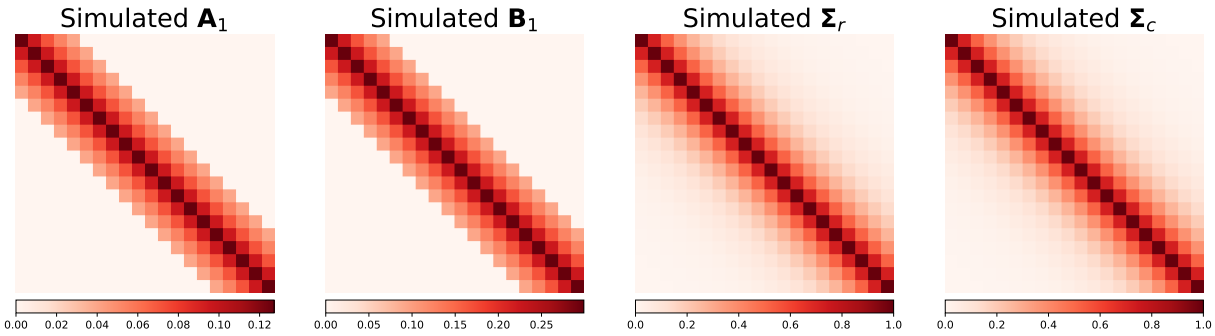


Figure C.1: Visualization of the simulated $\mathbf{A}_1, \mathbf{B}_1, \mathbf{\Sigma}_r, \mathbf{\Sigma}_c$ with $M = N = 20$.

To generate $g_1, g_2, g_3 \in \mathbb{H}_k$ and mimic the spatial grid in our real data application in Section 4.6, we specify the 2-D spatial grid with the two dimensions being latitude and longitude of points on a unit sphere $\mathbb{S}^2$. Each of the evenly spaced $M \times N$ grid points has its polar-azimuthal coordinate pair as $(\theta_i, \phi_j) \in [0°, 180°] \times [0°, 360°], i \in [M], j \in [N]$, and one projects the sampled grid points on the sphere onto a plane to form an $M \times N$ matrix. The polar θ (co-latitude) and azimuthal ϕ (longitude) angles are very commonly used in the spherical coordinate system, with the corresponding Euclidean coordinates being $(x, y, z) = (\sin(\theta)\cos(\phi), \sin(\theta)\sin(\phi), \cos(\theta))$.

As for the spatial kernel, we choose the Lebedev kernel:

$$k_\eta(s_1, s_2) = \left(\frac{1}{4\pi} + \frac{\eta}{12\pi}\right) - \frac{\eta}{8\pi}\sqrt{\frac{1 - \langle s_1, s_2\rangle}{2}}, \quad s_1, s_2 \in \mathbb{S}^2, \tag{C.55}$$

where $\langle \cdot, \cdot \rangle$ denotes the angle between two points on the sphere $\mathbb{S}^2$ and η is a hyperparameter of the kernel. In the simulation experiment as well as the real data application, we fix $\eta = 3$. The Lebedev kernel has the spherical harmonics functions as its eigenfunction:

$$k_\eta(s_1, s_2) = \frac{1}{4\pi} + \sum_{l=1}^{\infty} \frac{\eta}{(4l^2 - 1)(2l + 3)} \sum_{m=-l}^{l} Y_l^m(s_1) Y_l^m(s_2),$$

where $Y_l^m(\cdot)$ is a series of orthonormal real spherical harmonics bases defined on sphere $\mathbb{S}^2$:

$$Y_l^m(s) = Y_l^m(\theta, \phi) = \begin{cases} \sqrt{2} N_{lm} P_l^m(\cos(\theta)) \cos(m\phi) & \text{if } m > 0, \\ N_{l0} P_l^0(\cos(\theta)) & \text{if } m = 0, \\ \sqrt{2} N_{l|m|} P_l^{|m|}(\cos(\theta)) \sin(|m|\phi) & \text{if } m < 0, \end{cases}$$

with $N_{lm} = \sqrt{(2l + 1)(l - m)!/(4\pi(l + m)!)}$, and $P_l^m(\cdot)$ being the associated Legendre polynomials of order l. We refer our readers to Kennedy et al. (2013) for detailed information

about the spherical harmonics functions and the associated isotropic kernels. Under our 2-D grid setup and the choice of kernel, we have found that empirically, the kernel Gram matrix $\mathbf{K}$ has its eigen spectrum decaying at a rate at $\rho_i(\mathbf{K}) \approx i^{-r}$ with $r \in [1.3, 1.5]$.

We randomly sample g_1, g_2, g_3 from Gaussian processes with covariance kernel being the Lebedev kernel in (C.55). Finally, we simulate the vector time series $\mathbf{z}_t$ using a VAR(1) process. In Figure C.2, we visualize the simulated functional parameters as well as the vector time series from one random draw.

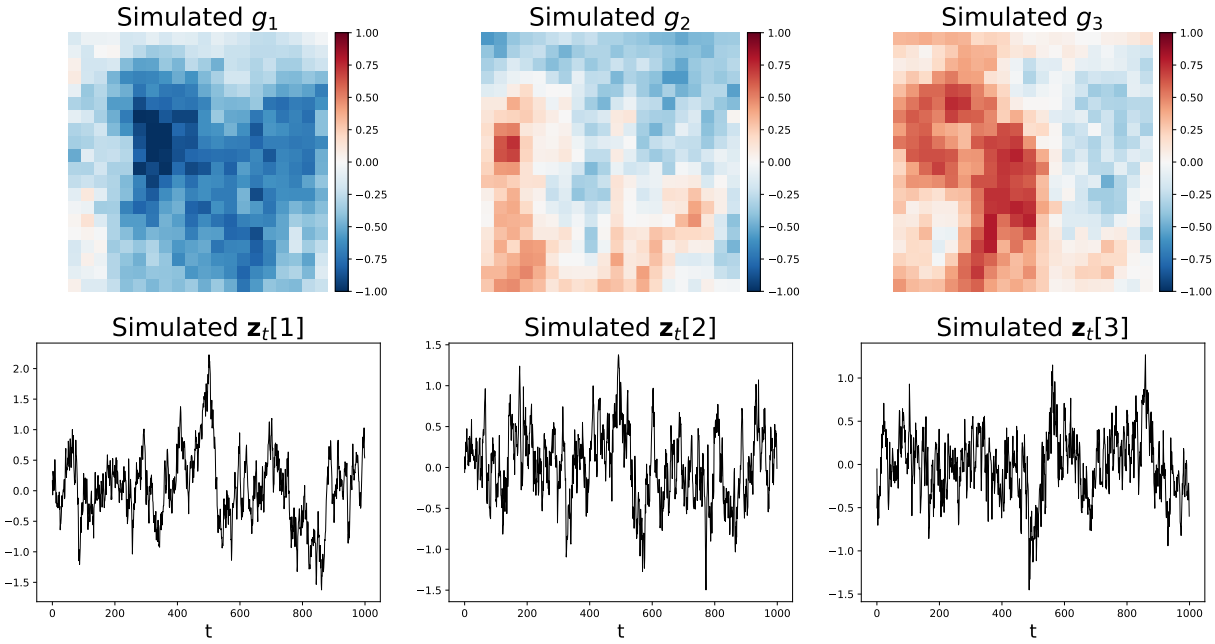


Figure C.2: Simulated functional parameters g_1, g_2, g_3 evaluated on a 20×20 spatial grid (top row) and the corresponding auxiliary vector time series (bottom row).

C.6.2 Approximated Penalized MLE with Kernel Truncation

The iterative algorithm in Section 4.3.1 requires inverting an $MND \times MND$ matrix in (4.15) when updating $\boldsymbol{\gamma}_q$, i.e., the coefficients of the representer functions $k(\cdot, s)$. One way to reduce the computational complexity without any approximation is to divide the step of updating $\boldsymbol{\gamma}_q = [\boldsymbol{\gamma}_{q,1}^\top : \cdots : \boldsymbol{\gamma}_{q,D}^\top]^\top$ to updating one block of parameters at a time following the order of $\boldsymbol{\gamma}_{q,1} \to \cdots \to \boldsymbol{\gamma}_{q,D}$. However, such a procedure requires inverting a matrix of size $MN \times MN$, which could still be high-dimensional.

To circumvent the issue of inverting large matrices, we can approximate the linear combination of all MN representers using a set of $R << MN$ basis functions, i.e., $\mathbf{K}\boldsymbol{\gamma}_{q,d} \approx \mathbf{K}_R \boldsymbol{\theta}_{q,d}$, where $\mathbf{K}_R \in \mathbb{R}^{MN \times R}, \boldsymbol{\theta}_{q,d} \in \mathbb{R}^R$. For example, one can reduce the spa-

tial resolution by subsampling a fraction of the rows and columns of the matrix and only use the representers at the subsampled "knots" as the basis functions. In this subsection, we consider an alternative approach by truncating the Mercer decomposition in (4.8).

Given the eigen-decomposition of $k(\cdot,\cdot)$ in (4.8), one can truncate the decomposition at the R^{th} largest eigenvalue λ_R and get an approximation: $k(\cdot,\cdot) \approx \sum_{r\le R} \lambda_r \psi_r(\cdot)\psi_r(\cdot)$. We will use the set of eigen-functions $\{\psi_1(\cdot),\ldots,\psi_R(\cdot)\}$ for faster computation. The choice of R depends on the decaying rate of the eigenvalue sequence $\{\lambda_r\}_{r=1}^{\infty}$ (thus the smoothness of the underlying functional parameters). Our simulation result shows that the estimation and prediction errors shrink monotonically as $R \to \infty$. Therefore, R can be chosen based on the computational resources available. The kernel truncation speeds up the computation at the cost of providing an overly-smoothed estimator, as we demonstrate next.

Given the kernel truncation, any functional parameter $g_{q,d}(\cdot)$ is now approximated as: $g_{q,d}(\cdot) \approx \sum_{r\in[R]} [\boldsymbol{\theta}_{q,d}]_r \psi_r(\cdot)$. The parameter to be estimated now is $\boldsymbol{\Theta}_q = [\boldsymbol{\theta}_{q,1};\cdots;\boldsymbol{\theta}_{q,D}] \in \mathbb{R}^{R\times D}$, whose dimension is much lower than before ($\boldsymbol{\Gamma}_q \in \mathbb{R}^{MN\times D}$). Estimating $\mathbf{vec}(\boldsymbol{\Theta}_q) = \boldsymbol{\theta}_q$ requires solving a ridge regression problem, and the updating formula is:

$$\boldsymbol{\theta}_q^{(l+1)} \leftarrow \left[\boldsymbol{\Phi}\left(\mathbf{z}_{t-q}^{\top}\otimes \mathbf{K}_R, \boldsymbol{\Sigma}^{(l)}\right) + \lambda T\left(\mathbf{I}_D \otimes \boldsymbol{\Lambda}_R^{-1}\right)\right]^{-1} \boldsymbol{\Phi}\left(\mathbf{z}_{t-q}^{\top}\otimes \mathbf{K}_R, \widetilde{\mathbf{x}}_{t,-q}, \boldsymbol{\Sigma}^{(l)}\right),$$

where $\mathbf{K}_R \in \mathbb{R}^{MN\times R}$ satisfies $[\mathbf{K}_R]_{ur} = \psi_r(s_{ij}), u = i+(j-1)M$, and $\boldsymbol{\Lambda}_r = \mathrm{diag}(\lambda_1,\ldots,\lambda_R)$, with λ_r being the r^{th} largest eigenvalue of the Mercer decomposition of $k(\cdot,\cdot)$. Now we only need to invert a matrix of size $RD \times RD$, which speeds up the computation.

In Figure C.3, we visualize the ground truth of g_3 and both its penalized MLE and truncated penalized MLE estimators. It is evident that the truncated penalized MLE estimators give a smooth approximation to g_3 and the approximation gets better when R gets larger. The choice of R should be as large as possible for accuracy, so one can determine R based on the computational resources available.

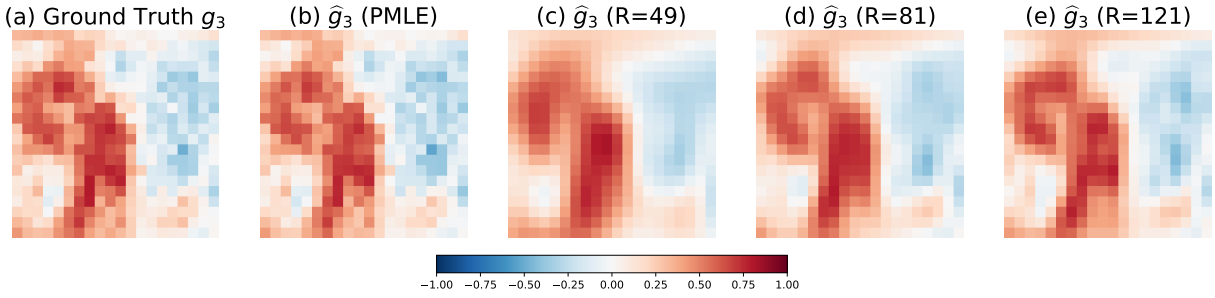


Figure C.3: Ground truth g_3 (panel (a)) against the penalized MLE estimator $\widehat{g}_3$ (panel (b)) and the truncated penalized MLE estimator $\widehat{g}_3$ using $R \in \{49, 81, 121\}$ basis functions. $M = 20$.

APPENDIX D

Appendix for Chapter 5

D.1 Proof of Lemma 5.2.1

The anisotropic total variation penalty can be simplified as follows, thanks to the rank-1 assumption on the feature map $\mathbf{W}_{s,t}$:

$$\begin{aligned}\|\mathbf{W}_{s,t}\|_{\text{TV}} &= \sum_{i=1}^{H-1}\sum_{j=1}^{W}|\mathbf{W}_{s,t}(i+1,j)-\mathbf{W}_{s,t}(i,j)| + \sum_{i=1}^{H}\sum_{j=1}^{W-1}|\mathbf{W}_{s,t}(i,j+1)-\mathbf{W}_{s,t}(i,j)| \\ &= \sum_{i=1}^{H-1}\sum_{j=1}^{W}|\mathbf{A}(s,i+1)-\mathbf{A}(s,i)|\cdot|\mathbf{B}(t,j)| \\ &+ \sum_{i=1}^{H}\sum_{j=1}^{W-1}|\mathbf{B}(t,j+1)-\mathbf{B}(t,j)|\cdot|\mathbf{A}(s,i)|.\end{aligned}$$

As a result, the total variation penalty has an elegant multiplicative formulation:

$$\begin{aligned}\lambda\sum_{s=1}^{h}\sum_{t=1}^{w}\|\mathbf{W}_{s,t}\|_{\text{TV}} &= \left(\lambda\sum_{t=1}^{w}\sum_{j=1}^{W}|\mathbf{B}(t,j)|\right)\cdot\left(\sum_{s=1}^{h}\sum_{i=1}^{H-1}|\mathbf{A}(s,i+1)-\mathbf{A}(s,i)|\right) \\ &+ \left(\lambda\sum_{t=1}^{w}\sum_{j=1}^{W-1}|\mathbf{B}(t,j+1)-\mathbf{B}(t,j)|\right)\cdot\left(\sum_{s=1}^{h}\sum_{i=1}^{H}|\mathbf{A}(s,i)|\right) \\ &= \lambda\cdot\|\mathbf{B}\|_1\cdot\|\nabla_x\mathbf{A}\|_1 + \lambda\cdot\|\nabla_x\mathbf{B}\|_1\cdot\|\mathbf{A}\|_1, \qquad \text{(D.1)}\end{aligned}$$

where ∇_x is the horizontal (i.e. row) first-order derivative operator and $\|\cdot\|_1$ is the matrix ℓ_1-norm. (D.1) turns out to be a fused-lasso type penalty (Tibshirani et al., 2005) with both a penalty on the sparsity of $\mathbf{A}$ and a penalty on the smoothness of each row of $\mathbf{A}$. This leads to a rank-1 feature map with row smoothness. Different from the fused-lasso penalty, we only introduce one tuning parameter λ instead of λ_1, λ_2 for the sparsity of smoothness of $\mathbf{A}$ separately. Instead, the tuning parameter for the sparsity and smoothness of $\mathbf{A}$ is

re-weighted by the smoothness and sparsity of $\mathbf{B}$ and vice versa, according to (D.1).

D.2 Proximal Gradient Descent for Tensor-GPST

Given the factorization assumption for the tensor multi-linear kernel factors $\mathbf{K}_1, \mathbf{K}_2, \mathbf{K}_3$:

$$\mathbf{K}_1 = \mathbf{U}_1^\top \mathbf{U}_1, \mathbf{K}_2 = \mathbf{U}_2^\top \mathbf{U}_2, \mathbf{K}_3 = \mathbf{U}_3^\top \mathbf{U}_3, \tag{D.2}$$

where $\mathbf{U}_1 \in \mathbb{R}^{r_1 \times h}, \mathbf{U}_2 \in \mathbb{R}^{r_2 \times w}, \mathbf{U}_3 \in \mathbb{R}^{r_3 \times C}$. One can rewrite the problem of minimizing the penalized negative log-likelihood as:

$$\begin{aligned}
\min_{\mathbf{A},\mathbf{B},\mathbf{U}_{1:3},\sigma} \quad & \frac{1}{2}\log\left|\widetilde{\mathbf{U}}\widetilde{\mathbf{U}}^\top + \mathbf{D}_\sigma\right| + \frac{1}{2}\mathbf{y}^\top\left(\widetilde{\mathbf{U}}\widetilde{\mathbf{U}}^\top + \mathbf{D}_\sigma\right)^{-1}\mathbf{y} \\
& + \lambda \sum_{s=1}^{h}\sum_{t=1}^{w}\Bigg\{\sum_{i=1}^{H-1}\sum_{j=1}^{W}|\mathbf{W}_{s,t}(i+1,j) - \mathbf{W}_{s,t}(i,j)| \\
& + \sum_{i=1}^{H}\sum_{j=1}^{W-1}|\mathbf{W}_{s,t}(i,j+1) - \mathbf{W}_{s,t}(i,j)|\Bigg\} \\
& = \ell(\mathbf{A},\mathbf{B},\mathbf{U}_{1:3},\sigma) + \lambda\sum_{s=1}^{h}\sum_{t=1}^{w}\|\mathbf{W}_{s,t}\|_{\text{TV}},
\end{aligned} \tag{D.3}$$

and recall that $\mathbf{y} = [y_1, y_2, \ldots, y_N]^\top$ and

$$\begin{aligned}
\widetilde{\boldsymbol{\mathcal{X}}} &= [\mathbf{vec}(\boldsymbol{\mathcal{X}}_1) : \mathbf{vec}(\boldsymbol{\mathcal{X}}_2) : \cdots : \mathbf{vec}(\boldsymbol{\mathcal{X}}_N)], \\
\widetilde{\mathbf{U}} &= \widetilde{\boldsymbol{\mathcal{X}}}^\top (\mathbf{I}_C \otimes \mathbf{B} \otimes \mathbf{A})^\top (\mathbf{U}_3 \otimes \mathbf{U}_2 \otimes \mathbf{U}_1)^\top.
\end{aligned}$$

We update the model parameters via a block coordinate descent scheme following the order of: $\mathbf{A} \to \mathbf{B} \to (\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3) \to \sigma \to \mathbf{A} \to \mathbf{B} \to \ldots$ until convergence.

The derivation of the gradients of $\ell(\cdot)$ w.r.t. $\mathbf{A}, \mathbf{B}, \mathbf{U}_{1:3}, \sigma$ have been made trivial thanks to the factorization assumption (D.2). For the $(i,j)^{\text{th}}$ element of $\mathbf{A}$, for instance, we have its partial derivative as:

$$\frac{\partial \ell}{\partial \mathbf{A}}(i,j) = \mathbf{tr}\left[\left(\frac{\partial \ell}{\partial \widetilde{\mathbf{U}}}\right)^\top \left(\frac{\partial \widetilde{\mathbf{U}}}{\partial \mathbf{A}(i,j)}\right)\right], \quad \frac{\partial \widetilde{\mathbf{U}}}{\partial \mathbf{A}(i,j)} = \widetilde{\boldsymbol{\mathcal{X}}}^\top (\mathbf{I}_C \otimes \mathbf{B} \otimes \mathbf{O}_{ij})^\top (\mathbf{U}_3 \otimes \mathbf{U}_2 \otimes \mathbf{U}_1)^\top,$$

where $\mathbf{O}_{ij} \in \mathbb{R}^{h \times H}$ is a binary matrix with all entries being zero except the $(i,j)^{\text{th}}$ entry

being one. The derivative of $\ell(\cdot)$ w.r.t. $\widetilde{\mathbf{U}}$ has an explicit form (Yu et al., 2018):

$$\frac{\partial \ell}{\partial \widetilde{\mathbf{U}}} = \widetilde{\mathbf{U}} \left(\mathbf{\Sigma}^{-1} + \mathbf{\Sigma}^{-1} \widetilde{\mathbf{U}}^\top \mathbf{y} \mathbf{D}_\sigma^{-1} \mathbf{y}^\top \widetilde{\mathbf{U}} \mathbf{\Sigma}^{-1} \right) - \mathbf{y} \mathbf{D}_\sigma^{-1} \mathbf{y}^\top \widetilde{\mathbf{U}} \mathbf{\Sigma}^{-1},$$

where $\mathbf{\Sigma} = \mathbf{D}_\sigma + \widetilde{\mathbf{U}}^\top \widetilde{\mathbf{U}}$. The derivative of $\widetilde{\mathbf{U}}$ w.r.t. $\mathbf{A}, \mathbf{B}, \mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3$ can be readily derived by simply replacing each matrix parameter with a sparse binary matrix such as $\mathbf{O}_{ij}$ defined above. For example for $\mathbf{U}_2$, one has:

$$\frac{\partial \widetilde{\mathbf{U}}}{\partial \mathbf{U}_2(i,j)} = \widetilde{\boldsymbol{\mathcal{X}}}^\top \left(\mathbf{I}_C \otimes \mathbf{B} \otimes \mathbf{A} \right)^\top \left(\mathbf{U}_3 \otimes \mathbf{O}_{ij} \otimes \mathbf{U}_1 \right)^\top,$$

where $\mathbf{O}_{ij} \in \mathbb{R}^{r_2 \times w}$ and is sparse except the $(i,j)^{\text{th}}$ element being one. The gradients for $\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3$ can be used for parameter update, and for $\mathbf{A}$ and $\mathbf{B}$, we consider updating them via proximal gradient descent. For $\mathbf{A}$ at the $(i+1)^{\text{th}}$ iteration, for example, one applies the gradient descent first to get an estimator proposal: $\widehat{\mathbf{A}}^{(i+\frac{1}{2})} = \widehat{\mathbf{A}}^{(i)} - \eta_i \partial \ell / \partial \mathbf{A}$, and then solves the following optimization problem, which is commonly known as the proximal step:

$$\mathbf{prox}_{\text{TV}}(\widehat{\mathbf{A}}^{(i+\frac{1}{2})}) = \underset{\mathbf{A} \in \mathbb{R}^{h \times H}}{\arg\min} \left\{ \frac{1}{2\eta_i} \left\| \mathbf{A} - \widehat{\mathbf{A}}^{(i+\frac{1}{2})} \right\|_{\text{F}}^2 + \lambda \sum_{s=1}^{h} \sum_{t=1}^{w} \|\mathbf{W}_{s,t}\|_{\text{TV}} \right\}, \tag{D.4}$$

where η_i is the step size of the $(i+1)^{\text{th}}$ step.

Solving the proximal problem in (D.4) can be broken down into multiple parallel 1-D *fused lasso signal approximation* problem. According to Proposition 1 of Friedman et al. (2007), solving (D.4) can be further broken down into first solving h total variation denoising problem (Rudin et al., 1992):

$$\widetilde{\mathbf{A}}(s,:) \leftarrow \underset{\boldsymbol{\alpha} \in \mathbb{R}^H}{\arg\min} \frac{1}{2\eta_i} \left\| \boldsymbol{\alpha} - \widehat{\mathbf{A}}^{(i+\frac{1}{2})}(s,:) \right\|_{\text{F}}^2 + \lambda \cdot \|\mathbf{B}\|_1 \cdot \sum_{j=2}^{H} |\boldsymbol{\alpha}(j+1) - \boldsymbol{\alpha}(j)|, \quad s = 1, 2, \ldots, h. \tag{D.5}$$

Then one can apply a soft-thresholding operator $\mathcal{S}_{\lambda \|\nabla_x \mathbf{B}\|_1}(\cdot)$, element-wisely, to $\widetilde{\mathbf{A}}$ to obtain the solution for (D.4). The problem in (D.5) can be efficiently solved via the python implementation in `prox-TV` based on a fast Newton's method (Jiménez and Sra, 2011; Barbero and Sra, 2018). Similar technique can be applied to update $\mathbf{B}$, and therefore the final optimization algorithm consists of both a gradient descent step and a fused-lasso proximal step. A more general theoretical discussion on the total variation penalty over 1-D signals can be found in Tibshirani (2014).

The gradient of $\ell(\cdot)$ w.r.t. σ^2 can be easily derived as follows:

$$\begin{aligned}\frac{\partial \ell}{\partial \sigma^2} &= \mathbf{tr}\left[\left(\frac{\partial \ell}{\partial (\mathbf{K}+\mathbf{D}_\sigma)^{-1}}\right)^\top \left(\frac{\partial (\mathbf{K}+\mathbf{D}_\sigma)^{-1}}{\partial \sigma^2}\right)\right] \\ &= \mathbf{tr}\left[\frac{1}{2}(\mathbf{K}+\mathbf{D}_\sigma)^{-1} - \frac{1}{2}(\mathbf{K}+\mathbf{D}_\sigma)^{-2}\mathbf{y}\mathbf{y}^\top\right]. \end{aligned} \tag{D.6}$$

Predictions on the unseen testing data with covariates $\mathbf{X}_*$, given the training data $(\mathbf{X}, \mathbf{y})$, can be easily derived using the predictive distribution $(\mathbf{y}_*|\mathbf{X}_*, \mathbf{X}, \mathbf{y}) \sim \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$:

$$\begin{aligned}\boldsymbol{\mu}_* &= \widehat{\mathcal{K}}(\mathbf{X}_*, \mathbf{X})\left(\widehat{\mathcal{K}}(\mathbf{X}, \mathbf{X}) + \mathbf{D}_{\widehat{\sigma}}\right)^{-1}\mathbf{y}, \\ \boldsymbol{\Sigma}_* &= \widehat{\mathcal{K}}(\mathbf{X}_*, \mathbf{X}_*) + \mathbf{D}_{\widehat{\sigma}} - \widehat{\mathcal{K}}(\mathbf{X}_*, \mathbf{X})\left(\widehat{\mathcal{K}}(\mathbf{X}, \mathbf{X}) + \mathbf{D}_{\widehat{\sigma}}\right)^{-1}\widehat{\mathcal{K}}(\mathbf{X}_*, \mathbf{X})^\top, \end{aligned}$$

where $\widehat{\mathcal{K}}(\cdot, \cdot)$ is the kernel function in (5.7) but evaluated at the estimated model parameters, and $\widehat{\mathcal{K}}(\mathbf{X}_*, \mathbf{X})$ simply denotes the covariances between the unseen data $\mathbf{X}_*$ and the training data $\mathbf{X}$, and the other notations follow. Here the covariates $\mathbf{X}_*$ and $\mathbf{X}$ should be interpreted as the unseen tensor data and the collection of all training tensor data, respectively.

D.3 Proof of Theorem 5.2.2

The proof of Theorem 5.2.2 is largely based on the convergence results of proximal gradient descent but with additional consideration on the alternating descent scheme. We denote an arbitrary collection of model parameters as $\boldsymbol{\theta} := (\mathbf{A}, \mathbf{B}, \mathbf{U}_{1:3}, \sigma)$. Since we update the four blocks of parameters cyclically in Algorithm 5.1, within each iteration, we further denote the intermediate parameter updates as $\widehat{\boldsymbol{\theta}}^{(k)} \xrightarrow{\text{update A}} \widehat{\boldsymbol{\theta}}^{(k+\frac{1}{4})} \xrightarrow{\text{update B}} \widehat{\boldsymbol{\theta}}^{(k+\frac{1}{2})} \xrightarrow{\text{update U}_{1:3}} \widehat{\boldsymbol{\theta}}^{(k+\frac{3}{4})} \xrightarrow{\text{update } \sigma} \widehat{\boldsymbol{\theta}}^{(k+1)}$. In the remainder of the proof, we will use $\mathbf{U}$ to denote $\mathbf{U}_{1:3}$ for notational simplicity.

In order to show the upper bound of the difference of the loss function after K iterations with its global minimum $L(\boldsymbol{\theta}^*)$, we first show Lemma D.3.1:

Lemma D.3.1. *Given the alternating proximal gradient descent algorithm in Algorithm 5.1 and*

the assumptions made in Theorem 5.2.2, one can bound $L(\widehat{\boldsymbol{\theta}}^{(k+\frac{v}{4})}), v \in \{1,2,3,4\}$ *as:*

$$L(\widehat{\boldsymbol{\theta}}^{(k+\frac{1}{4})}) \leq L(\mathbf{A}^*, \widehat{\mathbf{B}}^{(k)}, \widehat{\mathbf{U}}^{(k)}, \widehat{\sigma}^{(k)}) + \frac{1}{2\alpha}\left\{\left\|\widehat{\mathbf{A}}^{(k)} - \mathbf{A}^*\right\|^2 - \left\|\widehat{\mathbf{A}}^{(k+1)} - \mathbf{A}^*\right\|^2\right\}, \tag{D.7}$$

$$L(\widehat{\boldsymbol{\theta}}^{(k+\frac{1}{2})}) \leq L(\widehat{\mathbf{A}}^{(k+1)}, \mathbf{B}^{(*)}, \widehat{\mathbf{U}}^{(k)}, \widehat{\sigma}^{(k)}) + \frac{1}{2\alpha}\left\{\left\|\widehat{\mathbf{B}}^{(k)} - \mathbf{B}^*\right\|^2 - \left\|\widehat{\mathbf{B}}^{(k+1)} - \mathbf{B}^*\right\|^2\right\}, \tag{D.8}$$

$$L(\widehat{\boldsymbol{\theta}}^{(k+\frac{3}{4})}) \leq L(\widehat{\mathbf{A}}^{(k+1)}, \widehat{\mathbf{B}}^{(k+1)}, \mathbf{U}^*, \widehat{\sigma}^{(k)}) + \frac{1}{2\alpha}\left\{\left\|\widehat{\mathbf{U}}^{(k)} - \mathbf{U}^*\right\|^2 - \left\|\widehat{\mathbf{U}}^{(k+1)} - \mathbf{U}^*\right\|^2\right\}, \tag{D.9}$$

$$L(\widehat{\boldsymbol{\theta}}^{(k+1)}) \leq L(\widehat{\mathbf{A}}^{(k+1)}, \widehat{\mathbf{B}}^{(k+1)}, \widehat{\mathbf{U}}^{(k+1)}, \sigma^*) + \frac{1}{2\alpha}\left\{\left\|\widehat{\sigma}^{(k)} - \sigma^*\right\|^2 - \left\|\widehat{\sigma}^{(k+1)} - \sigma^*\right\|^2\right\}, \tag{D.10}$$

where $\|\cdot\|$ *is the matrix Frobenious norm,* α *is a constant step size with* $\alpha \leq 1/\max\{M_{\mathbf{A}}, M_{\mathbf{B}}, M_{\mathbf{U}}, M_\sigma\}$ *and* $M_{\mathbf{A}}, M_{\mathbf{B}}, M_{\mathbf{U}}, M_\sigma$ *are the Lipchitz constants for* $\mathbf{A}, \mathbf{B}, \mathbf{U}, \sigma$ *for the gradient of* $\ell(\cdot)$*, i.e. the negative log-likelihood defined in* (5.10)*.*

Proof. It suffices to prove (D.7), and the rest of the bounds follow the same technique.

First, given that the gradient of $\ell(\cdot)$ w.r.t. to $\mathbf{A}$ is Lipschitz continuous with constant $M_{\mathbf{A}}$, i.e. $\|\nabla_{\mathbf{A}}\ell(\mathbf{A}_1) - \nabla_{\mathbf{A}}\ell(\mathbf{A}_2)\| \leq M_{\mathbf{A}}\|\mathbf{A}_1 - \mathbf{A}_2\|, \forall \mathbf{A}_1, \mathbf{A}_2$. Since the other parameters also share the same property but have different Lipschitz constant $M_{\mathbf{B}}, M_{\mathbf{U}}, M_\sigma$, we use $M \coloneqq \max\{M_{\mathbf{A}}, M_{\mathbf{B}}, M_{\mathbf{U}}, M_\sigma\}$ as the Lipschitz constant for all parameters. Given the Lipschitz continuity of the derivative, one has:

$$\ell(\widehat{\boldsymbol{\theta}}^{(k+\frac{1}{4})}) \leq \ell(\widehat{\boldsymbol{\theta}}^{(k)}) + \left\langle \nabla_{\mathbf{A}}\ell(\widehat{\boldsymbol{\theta}}^{(k)}), \widehat{\mathbf{A}}^{(k+1)} - \widehat{\mathbf{A}}^{(k)} \right\rangle + \frac{M}{2}\|\widehat{\mathbf{A}}^{(k+1)} - \widehat{\mathbf{A}}^{(k)}\|^2, \tag{D.11}$$

which is a direct result of the following inequality for any function $\ell(\cdot)$ with M-Lipschitz continuous derivative:

$$\ell(y) \leq \ell(x) + \langle \nabla_x \ell(x), y - x \rangle + \frac{M}{2}\|y - x\|^2.$$

Additionally, since $\ell(\cdot)$ is assumed as block-wise convex, one has a natural upper bound of $\ell(\widehat{\boldsymbol{\theta}}^{(k)})$ based on convexity:

$$\ell(\widehat{\boldsymbol{\theta}}^{(k)}) \leq \ell(\mathbf{A}^*, \widehat{\mathbf{B}}^{(k)}, \widehat{\mathbf{U}}^{(k)}, \widehat{\sigma}^{(k)}) - \left\langle \nabla_{\mathbf{A}}\ell(\widehat{\boldsymbol{\theta}}^{(k)}), \mathbf{A}^* - \widehat{\mathbf{A}}^{(k)} \right\rangle. \tag{D.12}$$

Combining (D.11) and (D.12), one obtains:

$$\ell(\widehat{\boldsymbol{\theta}}^{(k+\frac{1}{4})}) \leq \ell(\mathbf{A}^*, \widehat{\mathbf{B}}^{(k)}, \widehat{\mathbf{U}}^{(k)}, \widehat{\sigma}^{(k)}) + \left\langle \nabla_{\mathbf{A}}\ell(\widehat{\boldsymbol{\theta}}^{(k)}), \widehat{\mathbf{A}}^{(k+1)} - \mathbf{A}^* \right\rangle + \frac{M}{2}\|\widehat{\mathbf{A}}^{(k+1)} - \widehat{\mathbf{A}}^{(k)}\|^2. \tag{D.13}$$

Also, since $\widehat{\mathbf{A}}^{(k+1)}$ is obtained via a proximal step:

$$\widehat{\mathbf{A}}^{(k+1)} = \arg\min_{\mathbf{A}} \frac{1}{2\alpha} \left\| \mathbf{A} - \left(\widehat{\mathbf{A}}^{(k)} - \alpha \nabla_{\mathbf{A}} \ell(\widehat{\boldsymbol{\theta}}^{(k)}) \right) \right\|^2 + \lambda \mathrm{R}\left(\mathbf{A}, \widehat{\mathbf{B}}^{(k)} \right),$$

$\widehat{\mathbf{A}}^{(k+1)}$ should satisfy the following subgradient condition:

$$\mathrm{G}_\alpha(\widehat{\boldsymbol{\theta}}^{(k)}) - \nabla_{\mathbf{A}} \ell(\widehat{\boldsymbol{\theta}}^{(k)}) \in \lambda \cdot \partial_A \mathrm{R}\left(\widehat{\mathbf{A}}^{(k+1)}, \widehat{\mathbf{B}}^{(k)} \right), \tag{D.14}$$

where $\mathrm{G}_\alpha(\widehat{\boldsymbol{\theta}}^{(k)}) \coloneqq -\alpha^{-1}\left(\widehat{\mathbf{A}}^{(k+1)} - \widehat{\mathbf{A}}^{(k)} \right)$ is the proximal gradient. Using the definition of subgradient, one can achieve a trivial inequality as follows:

$$\lambda \mathrm{R}\left(\widehat{\mathbf{A}}^{(k+1)}, \widehat{\mathbf{B}}^{(k)} \right) + \left\langle \mathrm{G}_\alpha(\widehat{\boldsymbol{\theta}}^{(k)}) - \nabla_{\mathbf{A}} \ell(\widehat{\boldsymbol{\theta}}^{(k)}), \mathbf{A}^* - \widehat{\mathbf{A}}^{(k+1)} \right\rangle \leq \lambda \mathrm{R}\left(\mathbf{A}^*, \widehat{\mathbf{B}}^{(k)} \right). \tag{D.15}$$

Combining (D.13) and (D.15), we have:

$$\begin{aligned}
L(\widehat{\boldsymbol{\theta}}^{(k+\frac{1}{4})}) &= \ell(\widehat{\boldsymbol{\theta}}^{(k+\frac{1}{4})}) + \lambda \mathrm{R}\left(\widehat{\mathbf{A}}^{(k+1)}, \widehat{\mathbf{B}}^{(k)} \right) \\
&\leq L(\mathbf{A}^*, \widehat{\mathbf{B}}^{(k)}, \widehat{\mathbf{U}}^{(k)}, \widehat{\sigma}^{(k)}) + \left\langle \mathrm{G}_\alpha(\widehat{\boldsymbol{\theta}}^{(k)}), \widehat{\mathbf{A}}^{(k+1)} - \mathbf{A}^* \right\rangle + \frac{M}{2} \|\widehat{\mathbf{A}}^{(k+1)} - \widehat{\mathbf{A}}^{(k)}\|^2 \\
&\leq L(\mathbf{A}^*, \widehat{\mathbf{B}}^{(k)}, \widehat{\mathbf{U}}^{(k)}, \widehat{\sigma}^{(k)}) + \left\langle \mathrm{G}_\alpha(\widehat{\boldsymbol{\theta}}^{(k)}), \widehat{\mathbf{A}}^{(k)} - \alpha \mathrm{G}_\alpha(\widehat{\boldsymbol{\theta}}^{(k)}) - \mathbf{A}^* \right\rangle + \frac{1}{2\alpha} \left\| \alpha \mathrm{G}_\alpha(\widehat{\boldsymbol{\theta}}^{(k)}) \right\|^2 \\
&= L(\mathbf{A}^*, \widehat{\mathbf{B}}^{(k)}, \widehat{\mathbf{U}}^{(k)}, \widehat{\sigma}^{(k)}) + \frac{1}{2\alpha} \left\{ \left\| \widehat{\mathbf{A}}^{(k)} - \mathbf{A}^* \right\|^2 - \left\| \widehat{\mathbf{A}}^{(k+1)} - \mathbf{A}^* \right\|^2 \right\},
\end{aligned}$$

which completes the proof. □

In the classical proximal gradient descent context, where one updates a single parameter iteratively, the bound in Lemma D.3.1 leads to a convergence rate of the algorithm at $\mathcal{O}(1/K)$, after one adds up all the inequalities from iteration 1 to K. The key difference is that, on the right hand side of the inequality (D.7), the loss function is evaluated at the global minima of $\mathbf{A}$ and the value of $\mathbf{B}, \mathbf{U}, \sigma$ *at the* k^{th} *iteration*. We need to quantify the difference between $L(\mathbf{A}^*, \widehat{\mathbf{B}}^{(k)}, \widehat{\mathbf{U}}^{(k)}, \widehat{\sigma}^{(k)})$ and $L(\boldsymbol{\theta}^*)$ to reach the final error bound result. This result is given in Lemma D.3.2 below.

Lemma D.3.2. *The difference of* $L(\mathbf{A}^*, \widehat{\mathbf{B}}^{(k)}, \widehat{\mathbf{U}}^{(k)}, \widehat{\sigma}^{(k)})$ *and* $L(\boldsymbol{\theta}^*) \coloneqq L(\mathbf{A}^*, \mathbf{B}^*, \mathbf{U}^*, \sigma^*)$ *can be fully characterized as:*

$$L(\mathbf{A}^*, \widehat{\mathbf{B}}^{(k)}, \widehat{\mathbf{U}}^{(k)}, \widehat{\sigma}^{(k)}) - L(\boldsymbol{\theta}^*) \leq \frac{M}{2} \left\{ \|\widehat{\mathbf{B}}^{(k)} - \mathbf{B}^*\| + \|\widehat{\mathbf{U}}^{(k)} - \mathbf{U}^*\| + \|\widehat{\sigma}^{(k)} - \sigma^*\| \right\}^2 \tag{D.16}$$

$$+ \|\nabla_{\mathbf{B}} \ell(\boldsymbol{\theta}^*)\| \cdot \|\widehat{\mathbf{B}}^{(k)} - \mathbf{B}^*\| + \lambda \mathrm{R}\left(\mathbf{A}^*, \widehat{\mathbf{B}}^{(k)} - \mathbf{B}^* \right), \tag{D.17}$$

where (D.16) *is the additional loss incurred by using the iterative value of the other parameters instead of the global optimum* (*called the ALT-gap*)*, and* (D.17) *is the additional loss incurred by using the total variation penalty with the* k^{th} *iterative value of* $\mathbf{B}$ (*called the TV-gap*).

Proof. We start the derivation with the following trivial decomposition:

$$L(\mathbf{A}^*, \widehat{\mathbf{B}}^{(k)}, \widehat{\mathbf{U}}^{(k)}, \widehat{\sigma}^{(k)}) - L(\boldsymbol{\theta}^*) = L(\mathbf{A}^*, \widehat{\mathbf{B}}^{(k)}, \widehat{\mathbf{U}}^{(k)}, \widehat{\sigma}^{(k)}) - L(\mathbf{A}^*, \mathbf{B}^*, \widehat{\mathbf{U}}^{(k)}, \widehat{\sigma}^{(k)}) \tag{D.18}$$

$$+ L(\mathbf{A}^*, \mathbf{B}^*, \widehat{\mathbf{U}}^{(k)}, \widehat{\sigma}^{(k)}) - L(\mathbf{A}^*, \mathbf{B}^*, \mathbf{U}^*, \widehat{\sigma}^{(k)}) \tag{D.19}$$

$$+ L(\mathbf{A}^*, \mathbf{B}^*, \mathbf{U}^*, \widehat{\sigma}^{(k)}) - L(\boldsymbol{\theta}^*). \tag{D.20}$$

We need to bound (D.18), (D.19) and (D.20) separately. For (D.18), we have:

$$\begin{aligned}
&L(\mathbf{A}^*, \widehat{\mathbf{B}}^{(k)}, \widehat{\mathbf{U}}^{(k)}, \widehat{\sigma}^{(k)}) - L(\mathbf{A}^*, \mathbf{B}^*, \widehat{\mathbf{U}}^{(k)}, \widehat{\sigma}^{(k)}) \\
&\leq \ell(\mathbf{A}^*, \widehat{\mathbf{B}}^{(k)}, \widehat{\mathbf{U}}^{(k)}, \widehat{\sigma}^{(k)}) - \ell(\mathbf{A}^*, \mathbf{B}^*, \widehat{\mathbf{U}}^{(k)}, \widehat{\sigma}^{(k)}) + \lambda \mathrm{R}\left(\mathbf{A}^*, \widehat{\mathbf{B}}^{(k)} - \mathbf{B}^*\right) \\
&\leq \left\langle \nabla_{\mathbf{B}} \ell(\mathbf{A}^*, \mathbf{B}^*, \widehat{\mathbf{U}}^{(k)}, \widehat{\sigma}^{(k)}), \widehat{\mathbf{B}}^{(k)} - \mathbf{B}^* \right\rangle + \frac{M}{2} \|\widehat{\mathbf{B}}^{(k)} - \mathbf{B}^*\|^2 + \lambda \mathrm{R}\left(\mathbf{A}^*, \widehat{\mathbf{B}}^{(k)} - \mathbf{B}^*\right) \\
&\leq \|\widehat{\mathbf{B}}^{(k)} - \mathbf{B}^*\| \cdot \left(\|\nabla_{\mathbf{B}} \ell(\boldsymbol{\theta}^*)\| + M \|\widehat{\mathbf{U}}^{(k)} - \mathbf{U}^*\| + M |\widehat{\sigma}^{(k)} - \sigma^*| \right) \\
&+ \frac{M}{2} \|\widehat{\mathbf{B}}^{(k)} - \mathbf{B}^*\|^2 + \lambda \mathrm{R}\left(\mathbf{A}^*, \widehat{\mathbf{B}}^{(k)} - \mathbf{B}^*\right),
\end{aligned} \tag{D.21}$$

where the last line follows from the Cauchy-Schwartz inequality followed by the Lipschitz continuity of the gradient and the triangle inequality of the Frobenius norm.

As for (D.19), similar technique follows and lead to:

$$\begin{aligned}
&L(\mathbf{A}^*, \mathbf{B}^*, \widehat{\mathbf{U}}^{(k)}, \widehat{\sigma}^{(k)}) - L(\mathbf{A}^*, \mathbf{B}^*, \mathbf{U}^*, \widehat{\sigma}^{(k)}) \\
&= \ell(\mathbf{A}^*, \mathbf{B}^*, \widehat{\mathbf{U}}^{(k)}, \widehat{\sigma}^{(k)}) - \ell(\mathbf{A}^*, \mathbf{B}^*, \mathbf{U}^*, \widehat{\sigma}^{(k)}) \\
&\leq \left\langle \nabla_{\mathbf{U}} \ell(\mathbf{A}^*, \mathbf{B}^*, \mathbf{U}^*, \widehat{\sigma}^{(k)}), \widehat{\mathbf{U}}^{(k)} - \mathbf{U}^* \right\rangle + \frac{M}{2} \|\widehat{\mathbf{U}}^{(k)} - \mathbf{U}^*\|^2 \\
&\leq M \|\widehat{\mathbf{U}}^{(k)} - \mathbf{U}^*\| \cdot |\widehat{\sigma}^{(k)} - \sigma^*| + \frac{M}{2} \|\widehat{\mathbf{U}}^{(k)} - \mathbf{U}^*\|^2,
\end{aligned}$$

where the last line uses the fact that at the global optimum, we have $\nabla_{\mathbf{U}} \ell(\boldsymbol{\theta}^*) = 0$.

Similarly for (D.20), one has:

$$L(\mathbf{A}^*, \mathbf{B}^*, \mathbf{U}^*, \widehat{\sigma}^{(k)}) - L(\boldsymbol{\theta}^*) \leq \left\langle \nabla_{\sigma} \ell(\boldsymbol{\theta}^*), \widehat{\sigma}^{(k)} - \sigma^* \right\rangle + \frac{M}{2} |\widehat{\sigma}^{(k)} - \sigma^*|^2.$$

Combining the three individual upper bounds together yields the result and thereby completes the proof. □

Similar results in Lemma D.3.2 can be easily derived for $\mathbf{B}^*, \mathbf{U}^*, \sigma^*$. With these theoretical results, we are now ready to prove Theorem 5.2.2.

Proof. Combining the results in Lemma D.3.1 and D.3.2, we have the following upper bound for $L(\widehat{\boldsymbol{\theta}}^{(k+\frac{1}{4})}) - L(\boldsymbol{\theta}^*)$:

$$\begin{aligned} L(\widehat{\boldsymbol{\theta}}^{(k+\frac{1}{4})}) - L(\boldsymbol{\theta}^*) \leq \frac{M}{2}\left\{\|\widehat{\mathbf{B}}^{(k)} - \mathbf{B}^*\| + \|\widehat{\mathbf{U}}^{(k)} - \mathbf{U}^*\| + \|\widehat{\sigma}^{(k)} - \sigma^*\|\right\}^2 \\ + \|\nabla_{\mathbf{B}}\ell(\boldsymbol{\theta}^*)\| \cdot \|\widehat{\mathbf{B}}^{(k)} - \mathbf{B}^*\| + \lambda \mathrm{R}\left(\mathbf{A}^*, \widehat{\mathbf{B}}^{(k)} - \mathbf{B}^*\right) \\ + \frac{1}{2\alpha}\left\{\left\|\widehat{\mathbf{A}}^{(k)} - \mathbf{A}^*\right\|^2 - \left\|\widehat{\mathbf{A}}^{(k+1)} - \mathbf{A}^*\right\|^2\right\}. \end{aligned}$$

If $\mathbf{B}, \mathbf{U}, \sigma$ is fixed at $\mathbf{B}^*, \mathbf{U}^*, \sigma^*$ and one only updates $\mathbf{A}$ via the proximal gradient descent, the error bound here vanishes to its last term only and can be further reduced if one adds up the inequality from iteration 1 to K, leading to the classical proximal gradient descent convergence rate result. Similar bounds for $L(\widehat{\boldsymbol{\theta}}^{(k+\frac{1}{2})}), L(\widehat{\boldsymbol{\theta}}^{(k+\frac{3}{4})}), L(\widehat{\boldsymbol{\theta}}^{(k+1)})$ can be derived and we aggregate the results together as follows:

$$\begin{aligned} \sum_{k=0}^{K}\sum_{v=1}^{4}\left(L(\widehat{\boldsymbol{\theta}}^{(k+\frac{v}{4})}) - L(\boldsymbol{\theta}^*)\right) \leq \frac{1}{2\alpha}\left\{\|\widehat{\boldsymbol{\theta}}^{(0)} - \boldsymbol{\theta}^*\|^2 - \|\widehat{\boldsymbol{\theta}}^{(K)} - \boldsymbol{\theta}^*\|^2\right\} \\ + \sum_{k=0}^{K} h_\lambda(\widehat{\mathbf{A}}^{(k+1)} - \mathbf{A}^*, \widehat{\mathbf{B}}^{(k+1)} - \mathbf{B}^*, \widehat{\mathbf{B}}^{(k)} - \mathbf{B}^*) \\ + \frac{1}{2\alpha}\sum_{k=0}^{K}\tau\left(\widehat{\boldsymbol{\theta}}^{(k+1)}, \widehat{\boldsymbol{\theta}}^{(k)}, \boldsymbol{\theta}^*\right), \end{aligned} \tag{D.22}$$

where $h_\lambda(\widehat{\mathbf{A}}^{(k+1)} - \mathbf{A}^*, \widehat{\mathbf{B}}^{(k+1)} - \mathbf{B}^*, \widehat{\mathbf{B}}^{(k)} - \mathbf{B}^*)$ is the extra gap from the optimal loss created by the existence of the total variation penalty in the loss function (thus we call it the TV-gap) and is defined as:

$$\begin{aligned} & h_\lambda(\widehat{\mathbf{A}}^{(k+1)} - \mathbf{A}^*, \widehat{\mathbf{B}}^{(k+1)} - \mathbf{B}^*, \widehat{\mathbf{B}}^{(k)} - \mathbf{B}^*) \\ & = 3\|\nabla_{\mathbf{A}}\ell(\boldsymbol{\theta}^*)\| \cdot \|\widehat{\mathbf{A}}^{(k+1)} - \mathbf{A}^*\| + 2\|\nabla_{\mathbf{B}}\ell(\boldsymbol{\theta}^*)\| \cdot \|\widehat{\mathbf{B}}^{(k+1)} - \mathbf{B}^*\| \\ & + \|\nabla_{\mathbf{B}}\ell(\boldsymbol{\theta}^*)\| \cdot \|\widehat{\mathbf{B}}^{(k)} - \mathbf{B}^*\| + \lambda \mathrm{R}\left(\mathbf{A}^*, \widehat{\mathbf{B}}^{(k)} - \mathbf{B}^*\right) \\ & + 3\lambda \mathrm{R}\left(\widehat{\mathbf{A}}^{(k+1)} - \mathbf{A}^*, \mathbf{B}^*\right) + 2\lambda \mathrm{R}\left(\mathbf{A}^*, \widehat{\mathbf{B}}^{(k+1)} - \mathbf{B}^*\right) \\ & + 2\lambda \mathrm{R}\left(\widehat{\mathbf{A}}^{(k+1)} - \mathbf{A}^*, \widehat{\mathbf{B}}^{(k+1)} - \mathbf{B}^*\right), \end{aligned} \tag{D.23}$$

where $\mathrm{R}(\mathbf{A}, \mathbf{B})$ is the total variation penalty defined in Lemma 5.2.1. $\tau(\widehat{\boldsymbol{\theta}}^{(k+1)}, \widehat{\boldsymbol{\theta}}^{(k)}, \boldsymbol{\theta}^*)$ is

the extra gap from the optimal loss created by the usage of iterative value of the parameters during the alternating proximal gradient descent (thus we call it the ALT-gap) and is defined as:

$$\begin{aligned}\tau\left(\widehat{\boldsymbol{\theta}}^{(k+1)}, \widehat{\boldsymbol{\theta}}^{(k)}, \boldsymbol{\theta}^*\right) := &\left[\|\widehat{\mathbf{B}}^{(k)} - \mathbf{B}^*\| + \|\widehat{\mathbf{U}}^{(k)} - \mathbf{U}^*\| + \|\widehat{\sigma}^{(k)} - \sigma^*\|\right]^2 \\ &+ \left[\|\widehat{\mathbf{A}}^{(k+1)} - \mathbf{A}^*\| + \|\widehat{\mathbf{U}}^{(k)} - \mathbf{U}^*\| + \|\widehat{\sigma}^{(k)} - \sigma^*\|\right]^2 \\ &+ \left[\|\widehat{\mathbf{A}}^{(k+1)} - \mathbf{A}^*\| + \|\widehat{\mathbf{B}}^{(k+1)} - \mathbf{B}^*\| + \|\widehat{\sigma}^{(k)} - \sigma^*\|\right]^2 \\ &+ \left[\|\widehat{\mathbf{A}}^{(k+1)} - \mathbf{A}^*\| + \|\widehat{\mathbf{B}}^{(k+1)} - \mathbf{B}^*\| + \|\widehat{\mathbf{U}}^{(k+1)} - \mathbf{U}^*\|\right]^2. \end{aligned} \tag{D.24}$$

The final result can be derived from (D.22) by lower bounding the left-hand side:

$$\sum_{k=0}^{K}\sum_{v=1}^{4}\left(L(\widehat{\boldsymbol{\theta}}^{(k+\frac{v}{4})}) - L(\boldsymbol{\theta}^*)\right) \geq 4(K+1)\left(L(\widehat{\boldsymbol{\theta}}^{(K+1)}) - L(\boldsymbol{\theta}^*)\right),$$

which is evident given that each step is a descent step. □

Although we cannot fully verify the assumptions made, we plot the history of the loss function and the relative change of the model parameters for our real data application in Figure D.1. Empirically, our model demonstrates a convergence rate at $\mathcal{O}(1/K)$ (see the red curve fitted based on a polynomial model with function form $f(k) = a + b/(c+k)$).

D.4 Details of the Simulation Study

In this section we provide the details on generating the simulation data. Given the three types of data in Figure 5.2, we use two sparse and banded tensor contracting factors $\mathbf{A}^*, \mathbf{B}^*$ (see the top of Figure D.2a) to contract each channel to a 3×3 tensor (see Figure 5.1a about the contraction operation). $\mathbf{A}^*, \mathbf{B}^*$ in Figure D.2a essentially do a 5×5 block averaging for the four corners, four sides and the middle block of each channel data. So one can expect the Type 1 & 3 data in Figure 5.2 to have its *signal* in the four corners of the contracted tensor, and Type 2 has its *signal* in the middle block (see Figure 5.2 for an illustration). Given the contracted tensor, we use the multi-linear kernel, specified in Figure D.2a (bottom) to generate the response variables via:

$$\mathbf{y} \sim \mathcal{N}\left(\mathbf{0}_N, \mathbf{K}^* + \sigma^2\mathbf{I}_N\right),$$

where $\mathbf{K}^*(i,j) = \mathbf{vec}(\boldsymbol{\mathcal{X}}_i)^\top \left[\mathbf{K}_3^* \otimes \left(\mathbf{B}^\top\mathbf{K}_2^*\mathbf{B}\right) \otimes \left(\mathbf{A}^\top\mathbf{K}_1^*\mathbf{A}\right)\right] \mathbf{vec}(\boldsymbol{\mathcal{X}}_j)$ and $\sigma = 0.5$.

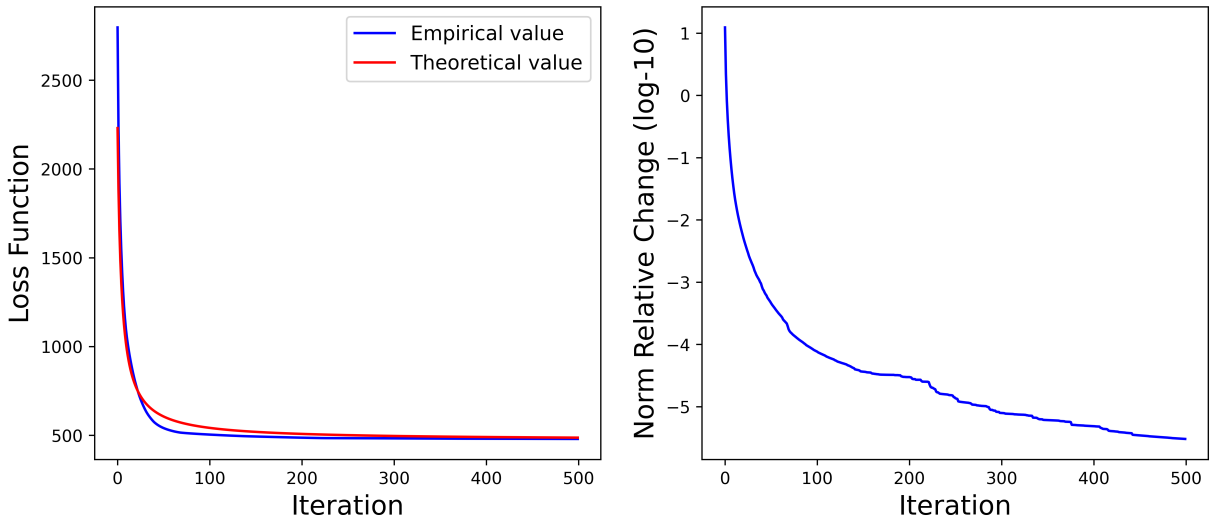


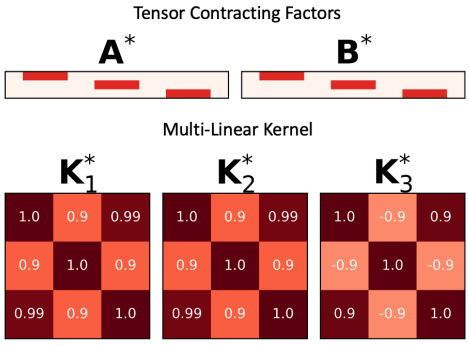
Figure D.1: (Left) Loss history of the solar flare intensity regression task with Tensor-GPST ($\lambda = 1.0$); A curve at the order of $\mathcal{O}(1/K)$ is fitted to the loss history and empirically, the algorithm converges at the rate of $\mathcal{O}(1/K)$ to a local minimum. (Right) History of the Frobenius norm of the relative change of model parameters in log-10 scale, which suggests that the parameters converge to a stationary point, and thus the ALT-gap and the TV-gap will converge to a constant.

One can notice from the kernel $\mathbf{K}_3^*$ in Figure D.2a that channel 1 & 3 are positively correlated and channel 2 is negatively correlated with both channel 1 & 3, and this is reflected in Figure D.2b, where we plot the distribution of the simulated sample of size $N = 500$, by the type of data. The tensor regression problem is to use the original $25 \times 25 \times 3$ tensors $\boldsymbol{\mathcal{X}}_i$ to forecast the regression label y_i.

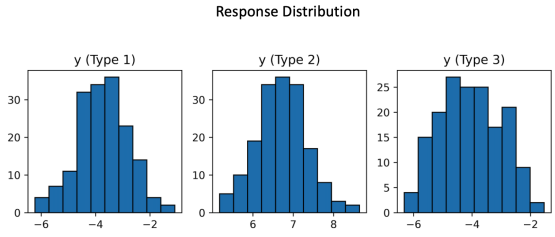
D.5 Details of the AIA-HMI Solar Imaging Dataset

In this appendix, we provide some astrophysical backgrounds and additional details on data preprocessing about the AIA-HMI solar flare imaging datasets.

There are over $12,000$ solar flares recorded by the Geostationary Operational Environmental Satellite (GOES) from May, 2010 to June, 2017, with intensity at least at the A-class flare level (peak X-ray brightness $< 10^{-7}\text{W/m}^2$). Among these flares, $4,409$ are B-class flares ($10^{-7} \sim 10^{-6}\text{W/m}^2$), 710 are M-class flares ($10^{-5} \sim 10^{-4}\text{W/m}^2$) and 50 are X-class flares ($> 10^{-4}\text{W/m}^2$). We combine the M-class and X-class flares in a single class, we name the class as M/X-class flares. Each flare is associated with a solar active region, which is a localized, transient volume of the solar atmosphere characterized by complex magnetic fields. We collect the AIA and HMI imaging data for each of the M/X-class flare during



(a) True tensor contracting factors (top) and true multi-linear kernels (bottom).



(b) Distribution of the response variable y by type (see type definition in Figure 5.2. Total sample size $N = 500$.

Figure D.2: Ground Truth of the Simulated data. (a) The true tensor contracting factors $(\mathbf{A}^*, \mathbf{B}^*)$ (top), where each has a banded structure with the 5 consecutive pixels filled with 0.2 on each row. The bottom shows the multi-linear kernel $\mathbf{K}_1^*, \mathbf{K}_2^*, \mathbf{K}_3^*$. (b) The resulting response distribution of each type of data. One can see how type 1 & 3 has similar distribution, thanks to their high channel correlation in $\mathbf{K}_3^*$.

this period, and collect the B-class flares happened within the same active regions to construct our own database. Given the data availability, we end up with a database of $1,264$ B-class flares and 728 M/X-class flares.

The AIA imaging data has 8 channels, distinguished by the wavelength band of the Extreme Ultraviolet (EUV) and Ultraviolet (UV) spectrum used to image the Sun [1]. The AIA channels are named under their respective spectral band: AIA-94Å, AIA-131Å, AIA-171Å, AIA-193Å, AIA-211Å, AIA-304Å, AIA-335Å and AIA-1600Å. The HMI imaging data captures the r, θ, ϕ-component of the solar magnetic field, and in our database, we keep the HMI $\mathrm{B_r}$ channel, which has demonstrated contains flare-predictive signals (Sun et al., 2022b). Finally, we derive the polarity inversion line (PIL) (Schrijver, 2007) from the $\mathrm{B_r}$, which highlights a sub-region with the strongest flare discriminating signals (Wang et al., 2020; Sun et al., 2021) and $\mathrm{B_r} \approx 0$. In Figure D.3, we plot one example of the 10 channels of HMI-AIA data for an M-class flare.

For the particular case in Figure D.3, the image size is 377×744, but different active regions are of different size. Also, different flares have their PIL, as well as the major signals in the other channels, stretching in different directions. To unify the size and orientation of all flares' imaging data, we follow these steps to preprocess our data:

- Pick the pixel in the PIL channel with the largest sum of PIL weights near its $51 \times$

[1] See more details at https://sdo.gsfc.nasa.gov/data/channels.php.

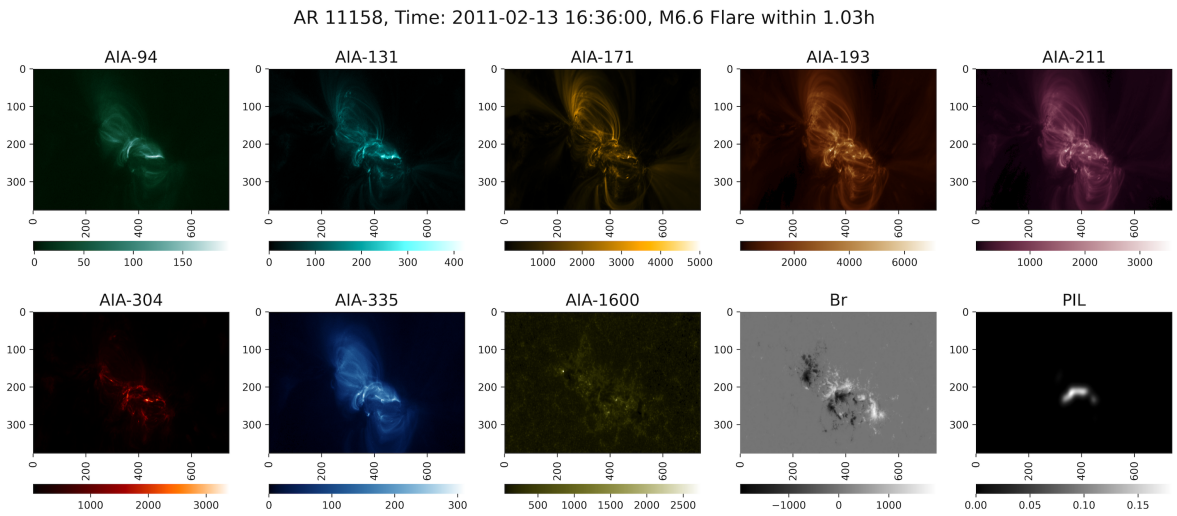


Figure D.3: M-class Flare Example for Active Region (AR) No.11158, recorded at 16:36:00 (UT) of Feb 13, 2021. The flare intensity is 6.6×10^{-5}W/m^2 and peaked at 17:38:00 (UT) of the same day. Tensor data size is $377 \times 744 \times 10$. Channel name labeled on top of each panel where we omit the Å.

51 neighborhood. This helps in picking the "center" of the image. If the PIL only contains zeros, which could happen for some very weak B-class flares, we use the AIA-1600Å in place of the PIL and follow the same procedure.

- Around the "center", we randomly sample $5,000$ pixels, with replacement, and the sampling probability is proportional to the PIL (or AIA-1600Å) pixel intensity, and do a Principal Component Analysis (PCA) of each pixel's 2D (x, y) coordinates (coordinates on the pixel grid) and use the first principal component to calculate the orientation of the PIL. This step helps to find the "direction" of the PIL.

- We rotate each channel with the same angle such that the "direction" of the PIL is vertical. Then, we crop a 201×201 window around the "center" of the image, and do zero-padding where it is needed.

These preprocessing steps create roughly comparable flare data across different active regions, but just as the simulation data pattern in Figure 5.2, there is still randomness w.r.t. the positioning and direction of the flare predictive signal for each individual flare. We subset our flare list to those whose longitude is within $\pm 60^\circ$ from the Sun's central meridian, which removes the low-quality samples with limb distortion. This reduces our sample size from $1,992$ flares to $1,329$ flares. We further reduce the dimensionality of the 201×201 images to 50×50, after applying the preprocessing steps above, by bi-linear interpolation to speed up the model computation. The preprocessed version of the sample

in Figure D.3, with tensor size $50 \times 50 \times 10$, is shown in Figure D.4. Notice how the PIL channel is now looking more "vertical" and how each channel is sort of "zoomed-in". The tensor size is now unified across all samples as $50 \times 50 \times 10$.

Before fitting the model, we normalize the scale of each channel such that each channel has its pixel intensity roughly within the range of $[-1, 1]$, to avoid numerical overflow in the algorithm. We only use the training set scale information to determine the scaling factor in order to avoid information spillover from the testing set.

For the flare intensity, originally, B-class flare has its intensity within $[10^{-7}, 10^{-6}]$ (unit: W/m^2), and M/X-class flare has its intensity within $[10^{-5}, +\infty]$ (unit: W/m^2). We transform any intensity y of each flare via:

$$\tilde{y} = \log_{10}(y) + 5.5,$$

such that the middle point of the weakest M/X-class flare and the strongest B-class flare is centered at zero.

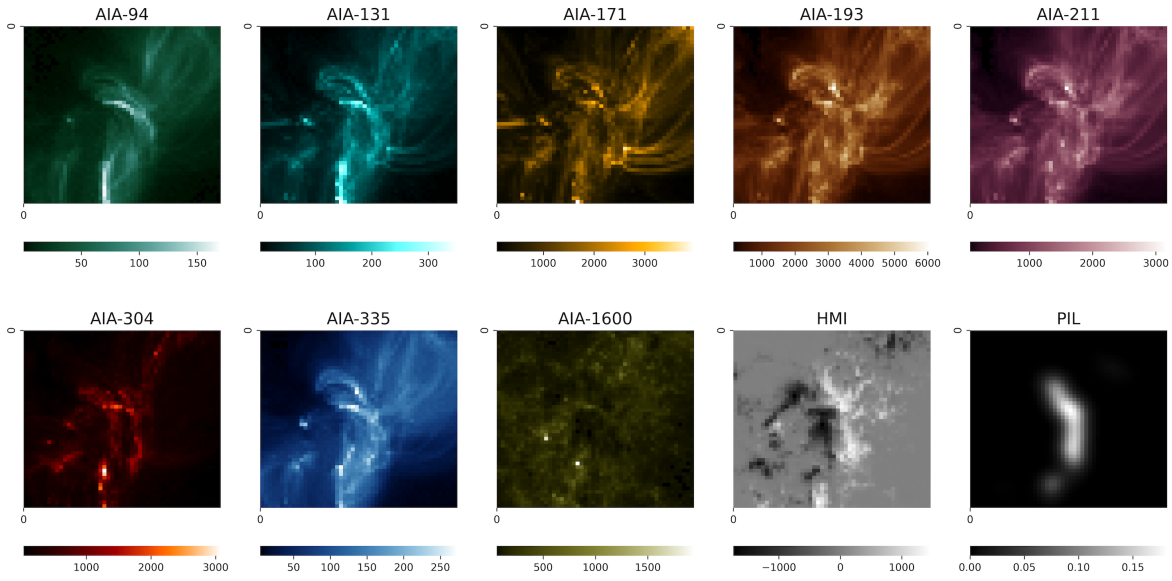


Figure D.4: Pre-processed version of the sample in Figure D.3. Notice how the PIL channel is now aligned vertically. Tensor size is reduced to $50 \times 50 \times 10$ for all $1,329$ flares.

D.6 Additional Results on Solar Flare Forecasting

This appendix provides additional results on the solar flare intensity regression. We first visualize the parameter estimates of **GPST**, which is the best-performing model in Ta-

ble 5.3, under one random train/test split with $\lambda = 1.0$ in this section. Figure D.5 provides the kernel estimates (the left three panels) and Figure D.6 shows the non-zero feature maps.

The kernel estimators $\widehat{\mathbf{K}}_1$ and $\widehat{\mathbf{K}}_2$ indicate that feature map $\mathbf{W}_{1,2}$ is of great importance since $\widehat{\mathbf{K}}_1(1,1)$ and $\widehat{\mathbf{K}}_2(2,2)$ contains the largest element, indicating that the feature extracted by $\mathbf{W}_{1,2}$ explains the most variations across all feature maps.

To formally conceptualize the notion of feature map importance as well as channel importance, one can start by decomposing the variations of the regression label y given tensor data $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{H\times W\times C}$ as follows:

$$\text{Var}(y) = \sum_{\substack{1\le s_1,s_2\le h\\ 1\le t_1,t_2\le w\\ 1\le c_1,c_2\le C}} \underbrace{\mathbf{K}_1(s_1,s_2)\cdot\mathbf{K}_2(t_1,t_2)}_{\text{Feature Map Importance}} \times \overbrace{\mathbf{K}_3(c_1,c_2)}^{\text{Channel Importance}} \times \underbrace{\left\langle \mathbf{W}_{s_1,t_1}, \mathbf{X}^{(c_1)}\right\rangle \cdot \left\langle \mathbf{W}_{s_2,t_2}, \mathbf{X}^{(c_2)}\right\rangle}_{\text{Latent Features Product}} + \overbrace{\sigma^2}^{\text{Noise}}, \tag{D.25}$$

and this leads to a natural definition of the percentage of explained variation for any *pair* of channels $c_1, c_2 \in [C]$:

$$\%\text{ Explained Variation} = \frac{\mathbf{K}_3(c_1,c_2)}{\text{Var}(y)} \times \sum_{\substack{1\le s_1,s_2\le h\\ 1\le t_1,t_2\le w}} \left\langle \mathbf{W}_{s_1,t_1}, \mathbf{X}^{(c_1)}\right\rangle \cdot \left\langle \mathbf{W}_{s_2,t_2}, \mathbf{X}^{(c_2)}\right\rangle \times \mathbf{K}_1(s_1,s_2)\cdot\mathbf{K}_2(t_1,t_2). \tag{D.26}$$

Similarly, one can define the percentage of explained variation for any *pair* of feature maps $\mathbf{W}_{s_1,t_1}, \mathbf{W}_{s_2,t_2}$, with $s_1, s_2 \in [h]$, $t_1, t_2 \in [w]$, as:

$$\%\text{ Explained Variation} = \frac{\mathbf{K}_1(s_1,s_2)\times\mathbf{K}_2(t_1,t_2)}{\text{Var}(y)} \times \sum_{1\le c_1,c_2\le C} \left\langle \mathbf{W}_{s_1,t_1}, \mathbf{X}^{(c_1)}\right\rangle \cdot \left\langle \mathbf{W}_{s_2,t_2}, \mathbf{X}^{(c_2)}\right\rangle \times \mathbf{K}_3(c_1,c_2). \tag{D.27}$$

The analysis here is a by-product of the Tensor-GPST model and is similar to the Joint and Individual Variation Explained (JIVE) (Lock et al., 2013) analysis. Both (D.26) and (D.27) can be computed empirically by plugging in the parameter estimators of $\widehat{\mathbf{K}}_1, \widehat{\mathbf{K}}_2, \widehat{\mathbf{K}}_3$ and $\widehat{\mathbf{W}}_{s,t}, s \in [h], t \in [w]$ and use all training inputs $\boldsymbol{\mathcal{X}}$ for calculation and take an average.

In the last two panels of Figure D.5, we show the percentage of explained variation for all 10 AIA-HMI channels based on (D.26) and the percentage of explained variation for all 9 feature maps based on (D.27). For channel-wise explained variation, we simply fix $c_1 = c_2$ in (D.26), and for feature map explained variation, we simply fix $(s_1, t_1) = (s_2, t_2)$. Note that since all channels share the same set of feature maps, the latent features

of different channels are not orthogonal, which indicates that the sum of the percentage of explained variation defined in (D.26) could exceed 100%. The same argument holds for the feature maps' explained variation. However, the explained variation still reveals the relative importance of different channels and feature maps.

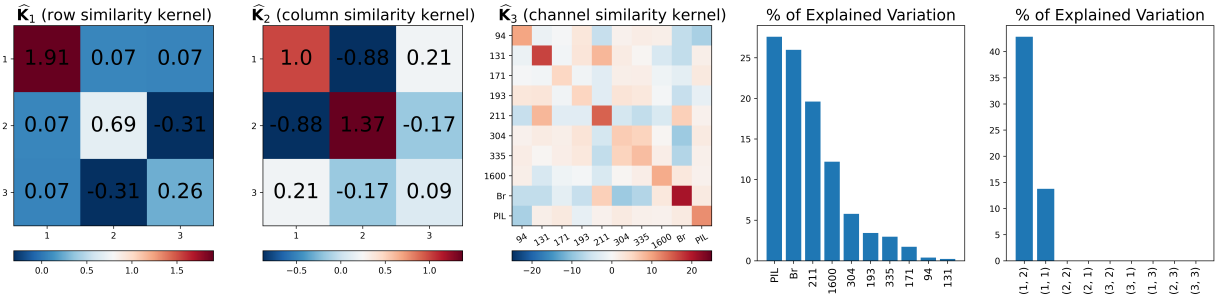


Figure D.5: **GPST** (under random train/test split, $\lambda = 1.0$) kernel estimates (panel 1-3), channel-wise $\%$ of explained variations (panel 4) and feature map $\%$ of explained variations (panel 5). It coincides with the literature (Wang et al., 2020; Sun et al., 2021) that the PIL is the channel with strong flare signals and the AIA imaging data is a good add-on to the HMI channel. The index for feature maps is the 2-tuple (s, t).

The feature maps shown in Figure D.6 mainly highlight two patterns:

- All six feature maps show non-zero weights in at least one of the four boundaries. This indicates that the features collected are around the perimeter of the flare eruptive region, which captures the "size" of the flare eruptive area. In Figure D.7 and D.8, we show the sample average of all 10 channels for the M/X-class and B-class flares, respectively. One can easily notice the difference between the two classes in terms of the "size" of the bright spots.

- There are some non-zero weights in $\mathbf{W}_{1,2}$ and other feature maps near row 20, where features are collected near the top of the brightest PIL region of the M/X flares.

Overall, the dimensionality that we set for the latent tensor is $3 \times 3 \times 10$, which is more than sufficient for summarizing the spatial data for each channel since only two features per channel have high feature importance. The feature importance for each channel further reveals that there are only 3-4 channels (e.g., PIL, Br, AIA-211Å, AIA-1600Å) that might be needed for summarizing the variations for the multi-channel imaging covariates across all samples. Our method is suggesting that the dimension reduction can be done further across the channels of the input data, which we leave for future work.

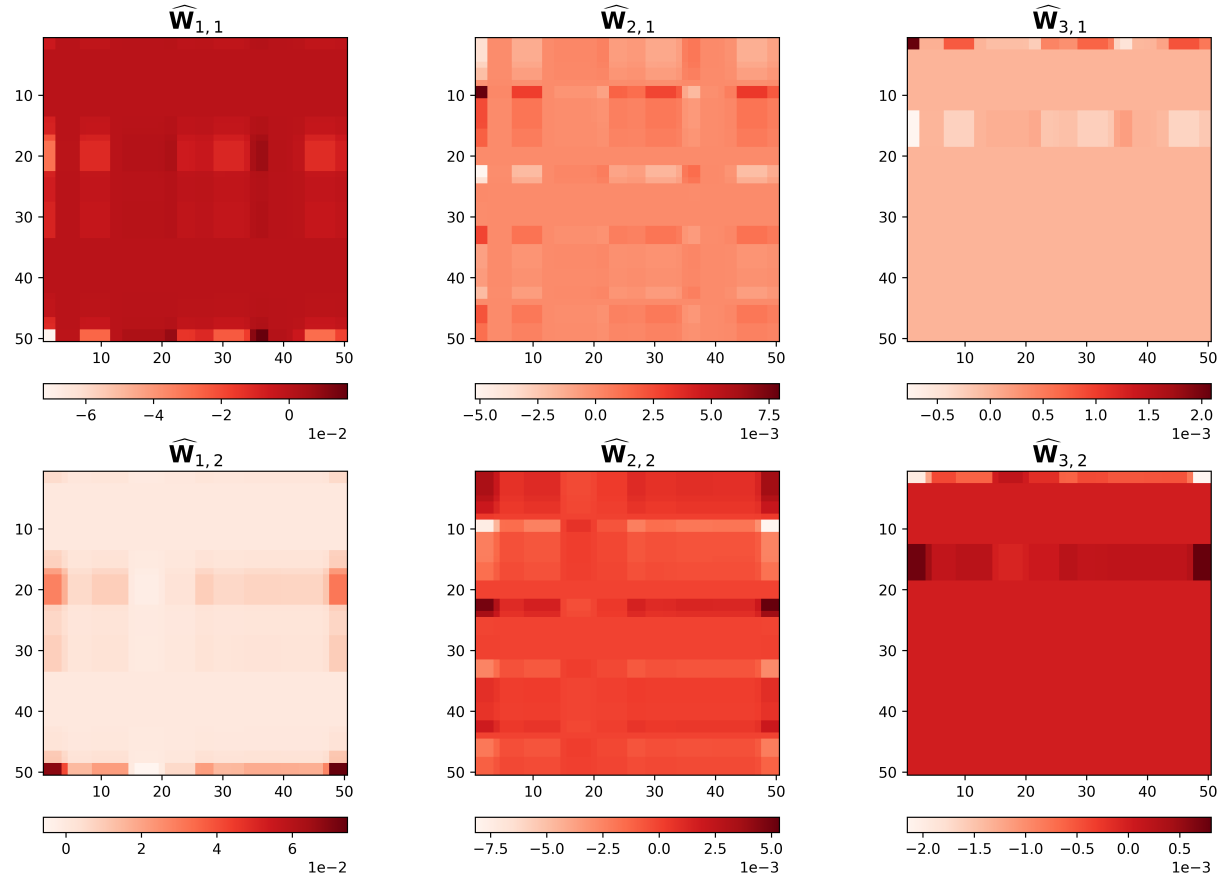


Figure D.6: **GPST** (under random train/test split, $\lambda = 1.0$) feature map (the non-zero ones) estimates.

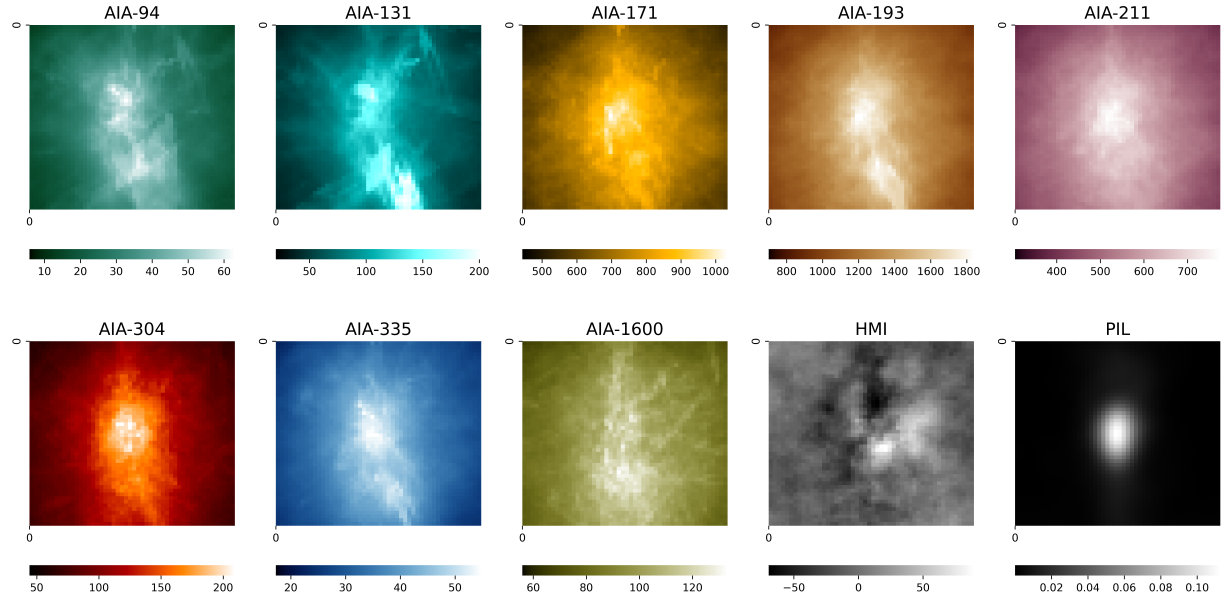


Figure D.7: Sample average AIA-HMI map for M-class flare.

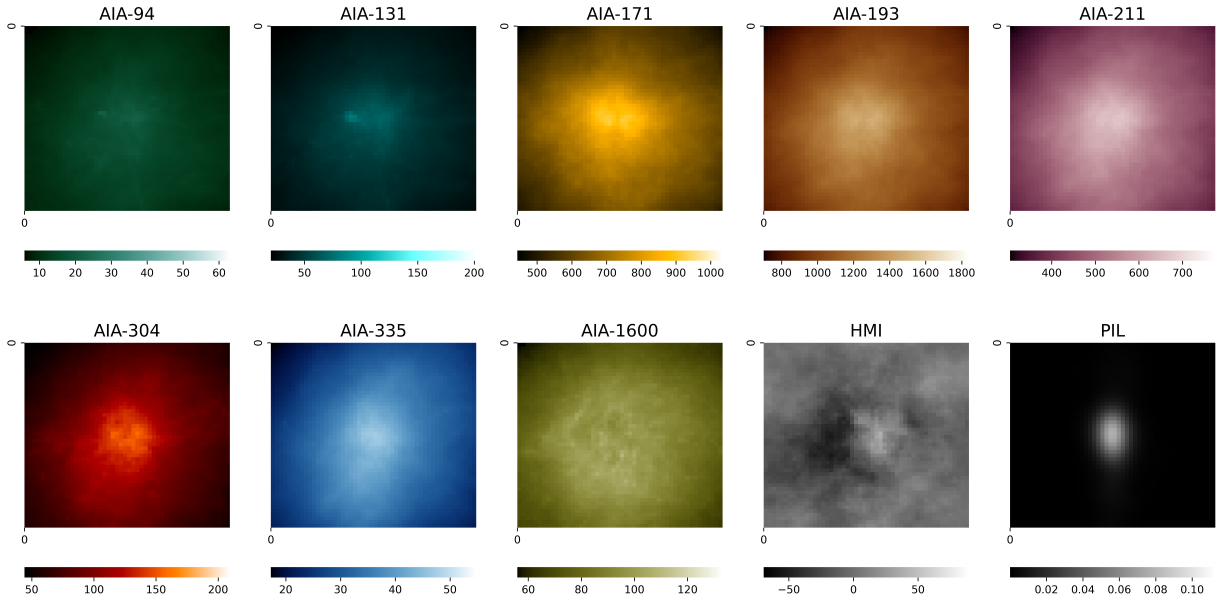


Figure D.8: Sample average AIA-HMI map for B-class flare.

BIBLIOGRAPHY

Ercha Aa, Wengeng Huang, Siqing Liu, Aaron Ridley, Shasha Zou, Liqin Shi, Yanhong Chen, Hua Shen, Tianjiao Yuan, Jianyong Li, et al. Midlatitude plasma bubbles over China and adjacent areas during a magnetic storm on 8 September 2017. Space Weather, 16(3):321–331, 2018.

Ercha Aa, Shasha Zou, Aaron Ridley, Shunrong Zhang, Anthea J. Coster, Philip J. Erickson, Siqing Liu, and Jiaen Ren. Merging of storm time midlatitude traveling ionospheric disturbances and equatorial plasma bubbles. Space Weather, 17(2):285–298, 2019.

Mangalathayil Ali Abdu. Day-to-day and short-term variabilities in the equatorial plasma bubble/spread F irregularity seeding and development. Progress in Earth and Planetary Science, 6(1):1–22, 2019.

Evrim Acar, Daniel M Dunlavy, Tamara G Kolda, and Morten Mørup. Scalable tensor factorizations for incomplete data. Chemometrics and Intelligent Laboratory Systems, 106(1):41–56, 2011.

H Akaike. Information theory and an extension of the maximum likelihood principle. In Proceedings of the 2nd International Symposium of Information Theory, pages 267–281. Akademiai Kiado, 1973.

Pierre Alquier. A Bayesian approach for noisy matrix completion: Optimal rate under general sampling distribution. Electronic Journal of Statistics, 9:823–841, 2015.

Hedy Attouch, Jérôme Bolte, and Benar Fux Svaiter. Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. Mathematical Programming, 137(1-2):91–129, 2013.

F Azpilicueta and B Nava. On the TEC bias of altimeter satellites. Journal of Geodesy, 95 (10):1–15, 2021.

Mohammad Taha Bahadori, Qi Rose Yu, and Yan Liu. Fast multivariate spatio-temporal analysis via low rank tensor learning. Advances in Neural Information Processing Systems, 27, 2014.

Rina Foygel Barber and Mathias Drton. High-dimensional Ising model selection with Bayesian Information Criteria. Electronic Journal of Statistics, 9:567–607, 2015.

Alvaro Barbero and Suvrit Sra. Modular proximal optimization for multidimensional total-variation regularization. Journal of Machine Learning Research, 19(1):2232–2313, 2018.

Sander Barendse. Efficiently weighted estimation of tail and interquantile expectations. preprint, 2022.

Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. ACM Transactions on Graphics, 28(3):24, 2009.

G. Barnes, K.D. Leka, E.A. Schumer, and D.J. Della-Rose. Probabilistic forecasting of solar flares from vector magnetogram data. Space Weather, 5(9), 2007.

S. Basu, K.M. Groves, Su. Basu, and P.J. Sultan. Specification and forecasting of scintillations in communication/navigation links: Current status and future plans. Journal of Atmospheric and Solar-Terrestrial Physics, 64(16):1745–1754, 2002.

Robert M Bell, Yehuda Koren, and Chris Volinsky. The bellkor solution to the Netflix prize. KorBell Team's Report to Netflix, 2007.

J. Bennett and S. Lanning. The Netflix prize. In Proceedings of the KDD Cup Workshop 2007, pages 3–6, New York, 2007. ACM.

Bhaswar B Bhattacharya and Sumit Mukherjee. Inference in Ising models. Bernoulli, 24 (1):493–525, 2018.

Xuan Bi, Annie Qu, and Xiaotong Shen. Multilayer tensor factorization with applications to recommender systems. The Annals of Statistics, 46(6B):3308 – 3333, 2018.

Christopher M Bishop and Nasser M Nasrabadi. Pattern recognition and machine learning, volume 4. Springer, 2006.

Space Studies Board. Space Weather: A Research Perspective. The National Academies Press, Washington, DC, 1997.

M. G. Bobra, X. Sun, J. T. Hoeksema, M. Turmon, Y. Liu, K. Hayashi, G. Barnes, and K. D. Leka. The Helioseismic and Magnetic Imager (HMI) vector magnetic field pipeline: SHARPs – Space-Weather HMI Active Region Patches. Solar Physics, 289(9):3549–3578, Sep 2014.

Monica G Bobra and Sebastien Couvidat. Solar flare prediction using SDO/HMI vector magnetic field data with a machine-learning algorithm. The Astrophysical Journal, 798 (2):135, 2015.

George EP Box and David R Cox. An analysis of transformations. Journal of the Royal Statistical Society Series B: Statistical Methodology, 26(2):211–243, 1964.

Mikio L Braun. Accurate error bounds for the eigenvalues of the kernel matrix. Journal of Machine Learning Research, 7:2303–2328, 2006.

Changxiao Cai, Gen Li, H Vincent Poor, and Yuxin Chen. Nonconvex low-rank tensor completion from noisy data. Operations Research, 70(2):1219–1237, 2022a.

Changxiao Cai, H Vincent Poor, and Yuxin Chen. Uncertainty quantification for nonconvex tensor completion: Confidence intervals, heteroscedasticity and optimality. IEEE Transactions on Information Theory, 69(1):407–452, 2022b.

Jian-Feng Cai, Jingyang Li, and Dong Xia. Generalized low-rank plus sparse tensor estimation by fast Riemannian optimization. Journal of the American Statistical Association, pages 1–17, 2022c.

Jian-Feng Cai, Jingyang Li, and Dong Xia. Provable tensor-train format tensor completion by Riemannian optimization. Journal of Machine Learning Research, 23(1):5365–5441, 2022d.

Jian-Feng Cai, Jingyang Li, and Dong Xia. Online tensor learning: Computational and statistical trade-offs, adaptivity and optimal regret. arXiv preprint arXiv:2306.03372, 2023.

T Tony Cai and Ming Yuan. Minimax and adaptive prediction for functional linear regression. Journal of the American Statistical Association, 107(499):1201–1216, 2012.

Emmanuel J Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. IEEE Transactions on Information Theory, 56(5):2053–2080, 2010.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In 2015 IEEE Symposium on Security and Privacy, pages 463–480. IEEE, 2015.

Alexandra Carpentier, Olga Klopp, Matthias Löffler, and Richard Nickl. Adaptive confidence sets for matrix completion. Bernoulli, 24(4A):2429–2460, 2018.

Alexandra Carpentier, Jens Eisert, David Gross, and Richard Nickl. Uncertainty quantification for matrix compressed sensing and quantum tomography problems. In High Dimensional Probability VIII: The Oaxaca Volume, pages 385–430. Springer, 2019.

Caihua Chen, Bingsheng He, and Xiaoming Yuan. Matrix completion via an alternating direction method. IMA Journal of Numerical Analysis, 32(1):227–245, 2012.

Elynn Y Chen, Dong Xia, Chencheng Cai, and Jianqing Fan. Semi-parametric tensor factor analysis by iteratively projected singular value decomposition. Journal of the Royal Statistical Society Series B: Statistical Methodology, 86(3):793–823, 2024.

Han Chen, Garvesh Raskutti, and Ming Yuan. Non-convex projected gradient descent for generalized low-rank tensor regression. Journal of Machine Learning Research, 20(1): 172–208, 2019a.

Rong Chen, Han Xiao, and Dan Yang. Autoregressive models for matrix-valued time series. Journal of Econometrics, 222(1):539–560, 2021a.

Yang Chen, Ward B Manchester, Alfred O Hero, Gabor Toth, Benoit DuFumier, Tian Zhou, Xiantong Wang, Haonan Zhu, Zeyu Sun, and Tamas I Gombosi. Identifying solar flare precursors using time series of SDO/HMI images and SHARP parameters. Space weather, 17(10):1404–1426, 2019b.

You-Lin Chen, Mladen Kolar, and Ruey S Tsay. Tensor canonical correlation analysis with convergence and statistical guarantees. Journal of Computational and Graphical Statistics, 30(3):728–744, 2021b.

Yuxin Chen, Jianqing Fan, Cong Ma, and Yuling Yan. Inference and uncertainty quantification for noisy matrix completion. Proceedings of the National Academy of Sciences, 116(46):22931–22937, 2019c.

Zhe Chen and Andrzej Cichocki. Nonnegative matrix factorization with temporal smoothness and/or spatial decorrelation constraints. Laboratory for Advanced Brain Signal Processing, RIKEN, Tech. Rep, 68, 2005.

Guang Cheng and Zuofeng Shang. Joint asymptotics for semi-nonparametric regression models with partially linear structure. The Annals of Statistics, 43:1351–1390, 2015.

Andrzej Cichocki, Danilo Mandic, Lieven De Lathauwer, Guoxu Zhou, Qibin Zhao, Cesar Caiafa, and Huy Anh Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. IEEE Signal Processing Magazine, 32 (2):145–163, 2015.

Barry A Cipra. An introduction to the Ising model. The American Mathematical Monthly, 94(10):937–959, 1987.

Robert S Conker, M Bakry El-Arini, Christopher J Hegarty, and Thomas Hsiao. Modeling the effects of ionospheric scintillation on GPS/satellite-based augmentation system availability. Radio Science, 38(1):1–1, 2003.

Noel Cressie. Kriging nonstationary data. Journal of the American Statistical Association, 81(395):625–634, 1986.

Noel Cressie and Gardar Johannesson. Fixed rank kriging for very large spatial data sets. Journal of the Royal Statistical Society Series B: Statistical Methodology, 70(1):209–226, 2008.

Noel Cressie and Christopher K Wikle. Statistics for spatio-temporal data. John Wiley & Sons, 2015.

Imre Csiszár and Zsolt Talata. Consistent estimation of the basic neighborhood of Markov random fields. The Annals of Statistics, 34(1):123–145, 2006.

Wenquan Cui, Haoyang Cheng, and Jiajing Sun. An RKHS-based approach to double-penalized regression in high-dimensional partially linear models. Journal of Multivariate Analysis, 168:201–210, 2018.

Abhirup Datta, Sudipto Banerjee, Andrew O Finley, and Alan E Gelfand. Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets. Journal of the American Statistical Association, 111(514):800–812, 2016.

Timo Dimitriadis and Sebastian Bayer. A joint quantile and expected shortfall regression framework. Electronic Journal of Statistics, 13:1823–1871, 2019.

Mingwang Dong, Linfu Huang, Xueqin Wu, and Qingguang Zeng. Application of least-squares method to time series analysis for 4DPM matrix. IOP Conference Series: Earth and Environmental Science, 455(1):012200, 2020.

Bradley Efron, Carl Morris, et al. Multivariate empirical Bayes and estimation of covariance matrices. The Annals of Statistics, 4(1):22–32, 1976.

D. A. Falconer, R. L. Moore, and G. A. Gary. Correlation of the coronal mass ejection productivity of solar active regions with measures of their global nonpotentiality from vector magnetograms: Baseline results. The Astrophysical Journal, 569:1016–1025, April 2002. doi: 10.1086/339161.

D. A. Falconer, R. L. Moore, and G. A. Gary. A measure from line-of-sight magnetograms for prediction of coronal mass ejections. Journal of Geophysical Research: Space Physics,, 108:1380, 2003.

D. A. Falconer, R. L. Moore, and G. A. Gary. Magnetic causes of solar coronal mass ejections: Dominance of the free magnetic energy over the magnetic twist alone. The Astrophysical Journal, 644:1258–1272, June 2006.

Vivek Farias, Andrew A Li, and Tianyi Peng. Uncertainty quantification for low-rank matrix completion with heterogeneous and sub-exponential noise. In International Conference on Artificial Intelligence and Statistics, pages 1179–1189. PMLR, 2022.

Maryam Fazel. Matrix rank minimization with applications. PhD thesis, Stanford University, 2002.

Maryam Fazel, Haitham Hindi, and Stephen P Boyd. A rank minimization heuristic with application to minimum order system approximation. In Proceedings of the 2001 American Control Conference, volume 6, pages 4734–4739. IEEE, 2001.

BK Fosdick and PD Hoff. Separable factor analysis with applications to mortality data. The Annals of Applied Statistics, 8(1):120–147, 2014.

JC Foster, AJ Coster, PJ Erickson, JM Holt, FD Lind, W Rideout, M McCready, A Van Eyken, RJ Barnes, RA Greenwald, et al. Multiradar observations of the polar tongue of ionization. Journal of Geophysical Research: Space Physics, 110(A9), 2005.

Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. Pathwise coordinate optimization. The Annals of Applied Statistics, 1(2):302–332, 2007.

Pere Giménez-Febrer, Alba Pagès-Zamora, and Georgios B Giannakis. Matrix completion and extrapolation via kernel regression. IEEE Transactions on Signal Processing, 67(19): 5004–5017, 2019.

Chong Gu. Smoothing Spline ANOVA models, 2nd edition. Springer, New York, 2013.

Leying Guan. Localized conformal prediction: A generalized inference framework for conformal prediction. Biometrika, 110(1):33–50, 2023.

Sharmistha Guha and Rajarshi Guhaniyogi. Bayesian generalized sparse symmetric tensor-on-vector regression. Technometrics, 63(2):160–170, 2021.

Rajarshi Guhaniyogi, Shaan Qamar, and David B Dunson. Bayesian tensor regression. Journal of Machine Learning Research, 18(1):2733–2763, 2017.

Yu Gui, Rina Barber, and Cong Ma. Conformalized matrix completion. Advances in Neural Information Processing Systems, 36:4820–4844, 2023.

Shaojun Guo, Yazhen Wang, and Qiwei Yao. High-dimensional and banded vector autoregressions. Biometrika, 103(4):889–903, 10 2016.

Weiwei Guo, Irene Kotsia, and Ioannis Patras. Tensor learning for regression. IEEE Transactions on Image Processing, 21(2):816–827, 2011.

Peter Hall and Christopher C Heyde. Martingale limit theory and its application. Academic press, 2014.

James D Hamilton. Time series analysis. Princeton University Press, 2020.

Rungang Han, Rebecca Willett, and Anru R Zhang. An optimal statistical and computational framework for generalized tensor estimation. The Annals of Statistics, 50(1):1–29, 2022.

Botao Hao, Boxiang Wang, Pengyuan Wang, Jingfei Zhang, Jian Yang, and Will Wei Sun. Sparse tensor additive regression. Journal of Machine Learning Research, 22, 2021.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edition. Springer, New York, 2009.

Trevor Hastie, Rahul Mazumder, Jason D Lee, and Reza Zadeh. Matrix completion and low-rank SVD via fast alternating least squares. Journal of Machine Learning Research, 16(1):3367–3402, 2015.

Xuming He, Kean Ming Tan, and Wen-Xin Zhou. Robust estimation and inference for expected shortfall regression with many regressors. Journal of the Royal Statistical Society Series B: Statistical Methodology, 85(4):1223–1246, 2023.

Manuel Hernández-Pajares, JM Juan, J Sanz, R Orus, A Garcia-Rigo, J Feltens, A Komjathy, SC Schaer, and A Krankowski. The IGS VTEC maps: A reliable source of ionospheric information since 1998. Journal of Geodesy, 83(3):263–275, 2009.

Manuel Hernández-Pajares, Haixia Lyu, Àngela Aragón-Àngel, Enric Monte-Moreno, Jingbin Liu, Jiachun An, and Hu Jiang. Polar electron content from GPS data-based global ionospheric maps: Assessment, case studies, and climatology. Journal of Geophysical Research: Space Physics, 125(6):e2019JA027677, 2020.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1):55–67, 1970.

Peter D Hoff. Separable covariance arrays via the tucker product, with applications to multivariate relational data. Bayesian Analysis, 6(2):179–196, 2011.

Sebastian Holtz, Thorsten Rohwedder, and Reinhold Schneider. On manifolds of tensors of fixed TT-rank. Numerische Mathematik, 120(4):701–731, 2012.

David Hong, Tamara G Kolda, and Jed A Duersch. Generalized canonical polyadic tensor decomposition. SIAM Review, 62(1):133–163, 2020.

The White House. National Space Weather Strategy and Action Plan, 2019.

Nan-Jung Hsu, Hsin-Cheng Huang, and Ruey S Tsay. Matrix autoregressive spatio-temporal models. Journal of Computational and Graphical Statistics, 30(4):1143–1155, 2021.

Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Image completion using planar structure guidance. ACM Transactions on Graphics, 33(4):1–10, 2014.

Hung Hung and Chen-Chien Wang. Matrix variate logistic regression model with application to EEG data. Biostatistics, 14(1):189–202, 2013.

Masaaki Imaizumi and Kohei Hayashi. Tensor decomposition with smoothness. In International Conference on Machine Learning, pages 1597–1606. PMLR, 2017.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. Advances in Neural Information Processing Systems, 28, 2015.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing, pages 665–674, 2013.

Chuanshu Ji and Lynne Seymour. A consistent model selection procedure for Markov random fields based on penalized pseudolikelihood. The Annals of Applied Probability, 6(2):423–443, 1996.

Zhenbang Jiao, Hu Sun, Xiantong Wang, Ward Manchester, Tamas Gombosi, Alfred Hero, and Yang Chen. Solar flare intensity prediction with machine learning models. Space Weather, 18(7):e2020SW002440, 2020.

Álvaro Barbero Jiménez and Suvrit Sra. Fast Newton-type methods for total variation regularization. In International Conference on Machine Learning, 2011.

Eric Jonas, Monica Bobra, Vaishaal Shankar, J Todd Hoeksema, and Benjamin Recht. Flare prediction using photospheric and coronal image data. Solar Physics, 293(3):48, 2018.

Jian Kang, Brian J Reich, and Ana-Maria Staicu. Scalar-on-image regression via the soft-thresholded Gaussian process. Biometrika, 105(1):165–184, 2018.

Esin Karahan, Pedro A Rojas-Lopez, Maria L Bringas-Vega, Pedro A Valdés-Hernández, and Pedro A Valdes-Sosa. Tensor analysis and fusion of multimodal brain images. Proceedings of the IEEE, 103(9):1531–1559, 2015.

Nikos Kargas, Cheng Qian, Nicholas D Sidiropoulos, Cao Xiao, Lucas M Glass, and Jimeng Sun. Stelar: Spatio-temporal tensor factorization with latent epidemiological regularization. In Proceedings of the AAAI Conference on Artificial Intelligence, pages 4830–4837, 2021.

Petr Kasalicky, Antoine Ledent, and Rodrigo Alves. Uncertainty-adjusted inductive matrix completion with graph neural networks. In Proceedings of the 17th ACM Conference on Recommender Systems, pages 1169–1174, 2023.

Matthias Katzfuss and Joseph Guinness. A general framework for Vecchia approximations of Gaussian processes. Statistical Science, 36(1):124–141, 2021.

Zheng Tracy Ke, Feng Shi, and Dong Xia. Community detection for hypergraph networks via regularized tensor power iteration. arXiv preprint arXiv:1909.06503, 2019.

Rodney A Kennedy, Parastoo Sadeghi, Zubair Khalid, and Jason D McEwen. Classification and construction of closed-form kernels for signal representation on the 2-sphere. In Wavelets and Sparsity XV, volume 8858, pages 169–183. SPIE, 2013.

Hyunsoo Kim and Haesun Park. Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. SIAM Journal on Matrix Analysis and Applications, 30(2):713–730, 2008a.

Jingu Kim and Haesun Park. Toward faster nonnegative matrix factorization: A new algorithm and comparisons. In 2008 Eighth IEEE International Conference on Data Mining, pages 353–362. IEEE, 2008b.

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. SIAM Review, 51(3):455–500, 2009.

Vladimir Koltchinskii and Evarist Giné. Random matrix approximation of spectra of integral operators. Bernoulli, pages 113–167, 2000.

Yehuda Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 426–434, 2008.

Yehuda Koren. The bellkor solution to the Netflix grand prize. Netflix prize documentation, 81(2009):1–10, 2009.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. Computer, 42(8):30–37, 2009.

Jean Kossaifi, Zachary C Lipton, Arinbjörn Kolbeinsson, Aran Khanna, Tommaso Furlanello, and Anima Anandkumar. Tensor regression networks. Journal of Machine Learning Research, 21(1):4862–4882, 2020.

Daniel Kressner, Michael Steinlechner, and Bart Vandereycken. Low-rank tensor completion by Riemannian optimization. BIT Numerical Mathematics, 54:447–468, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6):84–90, 2017.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 521(7553): 436–444, 2015.

Chanwoo Lee and Miaoyan Wang. Tensor denoising and completion based on ordinal observations. In International Conference on Machine Learning, pages 5778–5788. PMLR, 2020.

Daniel Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. Advances in Neural Information Processing Systems, 13:556–562, 2000.

Erich L Lehmann and George Casella. Theory of point estimation. Springer Science & Business Media, 2006.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. Journal of the American Statistical Association, 113(523):1094–1111, 2018.

James R. Lemen, Alan M. Title, David J. Akin, Paul F. Boerner, Catherine Chou, Jerry F. Drake, Dexter W. Duncan, Christopher G. Edwards, Frank M. Friedlaender, Gary F. Heyman, Neal E. Hurlburt, Noah L. Katz, Gary D. Kushner, Michael Levay, Russell W. Lindgren, Dnyanesh P. Mathur, Edward L. McFeaters, Sarah Mitchell, Roger A. Rehse, Carolus J. Schrijver, Larry A. Springer, Robert A. Stern, Theodore D. Tarbell, Jean-Pierre Wuelser, C. Jacob Wolfson, Carl Yanari, Jay A. Bookbinder, Peter N. Cheimets, David Caldwell, Edward E. Deluca, Richard Gates, Leon Golub, Sang Park, William A. Podgorski, Rock I. Bush, Philip H. Scherrer, Mark A. Gummin, Peter Smith, Gary Auker, Paul Jerram, Peter Pool, Regina Soufli, David L. Windt, Sarah Beardsley, Matthew Clapp, James Lang, and Nicholas Waltham. The Atmospheric Imaging Assembly (AIA) on the Solar Dynamics Observatory (SDO). Solar Physics, 275(1-2):17–40, January 2012.

Cai Li and Heping Zhang. Tensor quantile regression with application to association between neuroimages and human intelligence. The Annals of Applied Statistics, 15(3): 1455, 2021.

Lexin Li and Xin Zhang. Parsimonious tensor response regression. Journal of the American Statistical Association, 112(519):1131–1146, 2017.

Wei Li, Yinjian Wang, Na Liu, Chenchao Xiao, Zhiwei Sun, and Qian Du. Integrated spatio-spectral-temporal fusion via anisotropic sparsity constrained low-rank tensor approximation. IEEE Transactions on Geoscience and Remote Sensing, 2023.

Xiao Peng Li, Lei Huang, Hing Cheung So, and Bo Zhao. A survey on matrix completion: Perspective of signal processing. arXiv preprint arXiv:1901.10885, 2019.

Xiaoshan Li, Da Xu, Hua Zhou, and Lexin Li. Tucker tensor regression and neuroimaging analysis. Statistics in Biosciences, 10(3):520–545, 2018.

Zebang Li and Han Xiao. Multi-linear tensor autoregressive models. arXiv preprint arXiv:2110.00928, 2021.

Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(1):208–220, 2012.

Yipeng Liu, Jiani Liu, and Ce Zhu. Low-rank tensor train coefficient array estimation for tensor-on-tensor regression. IEEE Transactions on Neural Networks and Learning Systems, 31(12):5402–5411, 2020.

Zihuan Liu, Cheuk Yin Lee, and Heping Zhang. Tensor quantile regression with low-rank tensor train estimation. The Annals of Applied Statistics, 18(2):1294–1318, 2024.

Eric F Lock. Tensor-on-tensor regression. Journal of Computational and Graphical Statistics, 27(3):638–647, 2018.

Eric F Lock, Katherine A Hoadley, James Stephen Marron, and Andrew B Nobel. Joint and individual variation explained (JIVE) for integrated analysis of multiple data types. The Annals of Applied Statistics, 7(1):523, 2013.

Aurelie C Lozano, Hongfei Li, Alexandru Niculescu-Mizil, Yan Liu, Claudia Perlich, Jonathan Hosking, and Naoki Abe. Spatial-temporal causal modeling for climate change attribution. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 587–596, 2009.

Wenqi Lu, Zhongyi Zhu, and Heng Lian. High-dimensional quantile tensor regression. Journal of Machine Learning Research, 21(250):1–31, 2020.

Yuetian Luo and Anru R Zhang. Tensor-on-tensor regression: Riemannian optimization, over-parameterization, statistical-computational gap, and their interplay. arXiv preprint arXiv:2206.08756, 2022.

Huiying Mao, Ryan Martin, and Brian J Reich. Valid model-free spatial prediction. Journal of the American Statistical Association, pages 1–11, 2022.

Yun Mao and Lawrence K Saul. Modeling distances in large-scale networks by matrix factorization. In Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement, pages 278–287, 2004.

Takeru Matsuda, Masatoshi Uehara, and Aapo Hyvarinen. Information criteria for non-normalized models. Journal of Machine Learning Research, 22(158):1–33, 2021.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. Journal of Machine Learning Research, 11:2287–2322, 2010.

Michael Mendillo. Storms in the ionosphere: Patterns and processes for total electron content. Reviews of Geophysics, 44(4):RG4001, November 2006.

MIT Haystack Observatory. Madrigal database, 2012.

Naoto Nishizuka, Komei Sugiura, Yuki Kubo, Mitsue Den, and Mamoru Ishii. Deep flare net (DeFN) model for solar flare prediction. The Astrophysical Journal, 858(2):113, 2018.

Ivan V Oseledets. Tensor-train decomposition. SIAM Journal on Scientific Computing, 33 (5):2295–2317, 2011.

Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. Environmetrics, 5(2):111–126, 1994.

Yang Pan, Mingwu Jin, Shunrong Zhang, and Yue Deng. TEC map completion through a deep learning model: SNP-GAN. Space Weather, 19(11):e2021SW002810, 2021.

Georgia Papadogeorgou, Zhengwu Zhang, and David B Dunson. Soft tensor regression. Journal of Machine Learning Research, 22:219–1, 2021.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In Machine Learning: ECML 2002: 13th European Conference on Machine Learning Helsinki, Finland, August 19–23, 2002 Proceedings 13, pages 345–356. Springer, 2002.

Natalia E. Papitashvili and Joseph H. King. Omni 5-min Data [Data set]. NASA Space Physics Data Facility, 2020. https://doi.org/10.48322/gbpg-5r77.

Natasha Papitashvili, Dieter Bilitza, and Joseph King. OMNI: A description of near-Earth solar wind environment. 40th COSPAR Scientific Assembly, 40:C0–1, 2014.

Neal Parikh, Stephen Boyd, et al. Proximal algorithms. Foundations and Trends® in Optimization, 1(3):127–239, 2014.

Gerd W. Prölss. Ionospheric storms at mid-latitude: A short review. In Geophysical Monograph Series, volume 181, pages 9–24. American Geophysical Union, Washington, D. C., 2008.

Guillaume Rabusseau and Hachem Kadri. Low-rank regression with tensor responses. Advances in Neural Information Processing Systems, 29, 2016.

Pradeep Ravikumar, Martin J Wainwright, and John D Lafferty. High-dimensional Ising model selection using ℓ_1-regularized logistic regression. The Annals of Statistics, pages 1287–1319, 2010.

Benjamin Recht. A simpler approach to matrix completion. Journal of Machine Learning Research, 12(12), 2011.

William Rideout and Anthea Coster. Automated GPS processing for global total electron content data. GPS Solutions, 10(3):219–228, 2006.

David Roma-Dollase, Manuel Hernández-Pajares, Andrzej Krankowski, Kacper Kotulak, Reza Ghoddousi-Fard, Yunbin Yuan, Zishen Li, Hongping Zhang, Chuang Shi, Cheng Wang, et al. Consistency of seven different GNSS global ionospheric mapping techniques during one solar cycle. Journal of Geodesy, 92(6):691–706, 2018.

Samrat Roy and George Michailidis. Regularized high dimension low tubal-rank tensor regression. Electronic Journal of Statistics, 16(1):2683–2723, 2022.

Mark Rudelson and Roman Vershynin. Hanson-Wright inequality and sub-Gaussian concentration. Electronic Communications in Probability, pages 1–9, 2013.

Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena, 60(1-4):259–268, 1992.

Stefan Schaer. Mapping and predicting the Earth's ionosphere using the Global Positioning System. Dissertation of Astronomical Institute, page 205, 1999.

Stefan Schaer, Gerhard Beutler, Leos Mervart, Markus Rothacher, and Urs Wild. Global and regional ionosphere models using the GPS double difference phase observable. Proceeding of the IGS Workshop on Special Topics and New Directions, pages 77–92, 1995.

P. H. Scherrer, J. Schou, R. I. Bush, A. G. Kosovichev, R. S. Bogart, J. T. Hoeksema, Y. Liu, T. L. Duvall, J. Zhao, A. M. Title, C. J. Schrijver, T. D. Tarbell, and S. Tomczyk. The Helioseismic and Magnetic Imager (HMI) Investigation for the Solar Dynamics Observatory (SDO). Solar Physics, 275(1-2):207–227, January 2012.

Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In International Conference on Computational Learning Theory, pages 416–426. Springer, 2001.

Carolus J. Schrijver. A characteristic magnetic field pattern associated with all major solar flares and its use in flare forecasting. The Astrophysical Journal, 655(2):L117–L120, jan 2007.

Gideon Schwarz. Estimating the dimension of a model. The Annals of Statistics, 6(2): 461–464, 1978.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. Journal of Machine Learning Research, 9(3), 2008.

Zuofeng Shang and Guang Cheng. Local and global asymptotic inference in smoothing spline models. The Annals of Statistics, 41:2608–2638, 2013.

Zuofeng Shang and Guang Cheng. Nonparametric inference in generalized functional linear models. The Annals of Statistics, 43:1742–1773, 2015.

Meijia Shao and Yuan Zhang. Distribution-free matrix prediction under arbitrary missing pattern. arXiv preprint arXiv:2305.11640, 2023.

Bo Shen, Weijun Xie, and Zhenyu Kong. Smooth robust tensor completion for background/foreground separation with missing pixels: Novel algorithm with convergence guarantee. Journal of Machine Learning Research, 23(1):9757–9796, 2022.

Huanfeng Shen, Xiangchao Meng, and Liangpei Zhang. An integrated framework for the spatio–temporal–spectral fusion of remote sensing images. IEEE Transactions on Geoscience and Remote Sensing, 54(12):7135–7148, 2016.

S. G. Shepherd. Altitude-adjusted corrected geomagnetic coordinates: Definition and functional approximations. Journal of Geophysical Research: Space Physics, 119(9): 7501–7521, 2014.

Nathan Srebro, Jason Rennie, and Tommi S Jaakkola. Maximum-margin matrix factorization. In Advances in Neural Information Processing Systems, pages 1329–1336, 2005.

Michael Steinlechner. Riemannian optimization for high-dimensional tensor completion. SIAM Journal on Scientific Computing, 38(5):S461–S484, 2016.

James H Stock and Mark W Watson. Vector autoregressions. Journal of Economic perspectives, 15(4):101–115, 2001.

Jing Sui, Tülay Adali, Qingbao Yu, Jiayu Chen, and Vince D Calhoun. A review of multivariate methods for multimodal fusion of brain imaging data. Journal of Neuroscience Methods, 204(1):68–81, 2012.

Hu Sun and Yang Chen. Conformalized tensor completion with riemannian optimization. arXiv preprint arXiv:2405.00581, 2024.

Hu Sun, Ward Manchester IV, and Yang Chen. Improved and interpretable solar flare predictions with spatial and topological features of the polarity inversion line masked magnetograms. Space Weather, 19(12):e2021SW002837, 2021.

Hu Sun, Zhijun Hua, Jiaen Ren, Shasha Zou, Yuekai Sun, and Yang Chen. Matrix completion methods for the total electron content video reconstruction. The Annals of Applied Statistics, 16(3):1333–1358, 2022a.

Hu Sun, Yang Chen, Shasha Zou, Jiaen Ren, Yurui Chang, Zihan Wang, and Anthea Coster. Complete global total electron content map dataset based on a video imputation algorithm VISTA. Scientific Data, 10(1):236, 2023a.

Hu Sun, Ward Manchester, Meng Jin, Yang Liu, and Yang Chen. Tensor Gaussian process with contraction for multi-channel imaging analysis. In International Conference on Machine Learning, pages 32913–32935. PMLR, 2023b.

Hu Sun, Zuofeng Shang, and Yang Chen. Matrix autoregressive model with vector time series covariates for spatio-temporal data. arXiv preprint arXiv:2305.15671, 2023c.

Will Wei Sun and Lexin Li. STORE: Sparse tensor response regression and neuroimaging analysis. Journal of Machine Learning Research, 18(1):4908–4944, 2017.

Zeyu Sun, Monica G Bobra, Xiantong Wang, Yu Wang, Hu Sun, Tamas Gombosi, Yang Chen, and Alfred Hero. Predicting solar flares using CNN and LSTM on two solar cycles of active region data. The Astrophysical Journal, 931(2):163, 2022b.

Tiffany M Tang and Genevera I Allen. Integrated principal components analysis. Journal of Machine Learning Research, 22(198):1–71, 2021.

KF Tapping. The 10.7 cm solar radio flux (F10. 7). Space weather, 11(7):394–406, 2013.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. Journal of the Royal Statistical Society Series B: Statistical Methodology, 67(1):91–108, 2005.

Ryan J Tibshirani. Adaptive piecewise polynomial estimation via trend filtering. The Annals of Statistics, 42(1):285–323, 2014.

Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. Advances in Neural Information Processing Systems, 32, 2019.

Ryota Tomioka and Taiji Suzuki. Spectral norm of random tensors. arXiv preprint arXiv:1407.1870, 2014.

ShengLi Tzeng and Hsin-Cheng Huang. Resolution adaptive fixed rank kriging. Technometrics, 60(2):198–208, 2018.

Mark van der Wilk, Matthias Bauer, ST John, and James Hensman. Learning invariances using the marginal likelihood. Advances in Neural Information Processing Systems, 31, 2018.

JH van Zanten and Aad W van der Vaart. Reproducing kernel Hilbert spaces of Gaussian priors. In Pushing the Limits of Contemporary Statistics: Contributions in Honor of Jayanta K. Ghosh, pages 200–222. Institute of Mathematical Statistics, 2008.

Juha Vierinen, Anthea J. Coster, William C. Rideout, Philip J. Erickson, and Johannes Norberg. Statistical framework for estimating GNSS bias. Atmospheric Measurement Techniques, 9(3):1303–1312, 2016.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. Algorithmic Learning in a Random World, volume 29. Springer, 2005.

Di Wang, Yao Zheng, Heng Lian, and Guodong Li. High-dimensional vector autoregressive time series modeling via tensor decomposition. Journal of the American Statistical Association, 117(539):1338–1356, 2022.

Di Wang, Yao Zheng, and Guodong Li. High-dimensional low-rank tensor autoregressive time series modeling. Journal of Econometrics, 238(1):105544, 2024.

Dong Wang, Xialu Liu, and Rong Chen. Factor models for matrix-valued high-dimensional time series. Journal of Econometrics, 208(1):231–248, 2019.

Hailin Wang, Feng Zhang, Jianjun Wang, Tingwen Huang, Jianwen Huang, and Xinling Liu. Generalized nonconvex approach for low-tubal-rank tensor recovery. IEEE Transactions on Neural Networks and Learning Systems, 33(8):3305–3319, 2021a.

Hua Wang, Feiping Nie, and Heng Huang. Low-rank tensor completion with spatio-temporal consistency. In Proceedings of the AAAI Conference on Artificial Intelligence, pages 2846–2852, 2014.

Miaoyan Wang and Lexin Li. Learning from binary multiway data: Probabilistic tensor decomposition and its statistical optimality. Journal of Machine Learning Research, 21 (1):6146–6183, 2020.

Miaoyan Wang and Yuchen Zeng. Multiway clustering via tensor block models. Advances in Neural Information Processing Systems, 32, 2019.

Xiantong Wang, Yang Chen, Gabor Toth, Ward B Manchester, Tamas I Gombosi, Alfred O Hero, Zhenbang Jiao, Hu Sun, Meng Jin, and Yang Liu. Predicting solar flares with machine learning: Investigating solar cycle dependence. The Astrophysical Journal, 895(1):3, 2020.

Xiao Wang, Hongtu Zhu, and Alzheimer's Disease Neuroimaging Initiative. Generalized scalar-on-image regression models via total variation. Journal of the American Statistical Association, 112(519):1156–1168, 2017.

Zihan Wang, Shasha Zou, Lei Liu, Jiaen Ren, and Ercha Aa. Hemispheric asymmetries in the mid-latitude ionosphere during the September 7–8, 2017 storm: Multi-instrument observations. Journal of Geophysical Research: Space Physics, 126:e2020JA028829, 4 2021b.

Zihan Wang, Shasha Zou, Hu Sun, and Yang Chen. Forecast global ionospheric TEC: Apply modified U-Net on VISTA TEC data set. Space Weather, 21(8):e2023SW003494, 2023.

Bo Wei, Limin Peng, Ying Guo, Amita Manatunga, and Jennifer Stevens. Tensor response quantile regression with neuroimaging data. Biometrics, 79(3):1947–1958, 2023.

Christopher K Williams and Carl Edward Rasmussen. Gaussian Processes for Machine Learning, volume 2. MIT press Cambridge, MA, 2006.

Jia Wu, Xiu-Yun Chen, Hao Zhang, Li-Dong Xiong, Hang Lei, and Si-Hao Deng. Hyperparameter optimization for machine learning models based on Bayesian optimization. Journal of Electronic Science and Technology, 17(1):26–40, 2019.

Dong Xia and Ming Yuan. Statistical inferences of linear forms for noisy matrix completion. Journal of the Royal Statistical Society Series B: Statistical Methodology, 83(1):58–77, 2021.

Dong Xia, Ming Yuan, and Cun-Hui Zhang. Statistically optimal and computationally efficient low rank tensor completion from noisy entries. The Annals of Statistics, 49(1), 2021.

H Xiao, Y Han, R Chen, and C Liu. Reduced rank autoregressive models for matrix time series. Journal of Business and Economic Statistics, 2022.

Yangyang Xu and Wotao Yin. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. SIAM Journal on Imaging Sciences, 6(3):1758–1789, 2013.

Yangyang Xu, Ruru Hao, Wotao Yin, and Zhixun Su. Parallel matrix factorization for low-rank tensor completion. Inverse Problems & Imaging, 9(2):601–624, 2015.

Yun Yang, Zuofeng Shang, and Guang Cheng. Non-asymptotic analysis for nonparametric testing. In 33rd Annual Conference on Learning Theory, pages 1–47. ACM, 2020a.

Zhe Yang, YT Jade Morton, Irina Zakharenkova, Iurii Cherniak, Shuli Song, and Wei Li. Global view of ionospheric disturbance impacts on kinematic GPS positioning solutions during the 2015 St. Patrick's Day storm. Journal of Geophysical Research: Space Physics, 125(7):e2019JA027681, 2020b.

Waqar Younas, Majid Khan, C. Amory-Mazaudier, Paul O. Amaechi, and R. Fleury. Middle and low latitudes hemispheric asymmetries in $\Sigma O/N2$ and TEC during intense magnetic storms of solar cycle 24. Advances in Space Research, 69:220–235, 1 2022.

Rose Yu and Yan Liu. Learning from multiway data: Simple and efficient tensor regression. In International Conference on Machine Learning, pages 373–381. PMLR, 2016.

Rose Yu, Guangyu Li, and Yan Liu. Tensor regression meets Gaussian processes. In International Conference on Artificial Intelligence and Statistics, pages 482–490. PMLR, 2018.

Ming Yuan and T Tony Cai. A reproducing kernel Hilbert space approach to functional linear regression. The Annals of Statistics, 38(6):3412–3444, 2010.

Ming Yuan and Cun-Hui Zhang. On tensor completion via nuclear norm minimization. Foundations of Computational Mathematics, 16(4):1031–1068, 2016.

Henry Shaowu Yuchi, Simon Mak, and Yao Xie. Bayesian uncertainty quantification for low-rank matrix completion. Bayesian Analysis, 18(2):491–518, 2023.

Yoel Zeldes, Stavros Theodorakis, Efrat Solodnik, Aviv Rotman, Gil Chamiel, and Dan Friedman. Deep density networks and uncertainty in recommender systems. arXiv preprint arXiv:1711.02487, 2017.

Hongping Zhang, Peiliang Xu, Wenhui Han, Maorong Ge, and Chuang Shi. Eliminating negative VTEC in global ionosphere maps using inequality-constrained least squares. Advances in Space Research, 51(6):988–1000, 2013.

Shushu Zhang, Xuming He, Kean Ming Tan, and Wen-Xin Zhou. High-dimensional expected shortfall regression. arXiv preprint arXiv:2307.02695, 2023.

Xuchao Zhang, Liang Zhao, Arnold P Boedihardjo, Chang-Tien Lu, and Naren Ramakrishnan. Spatiotemporal event forecasting from incomplete hyper-local price data. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pages 507–516, 2017.

Hua Zhou and Lexin Li. Regularized matrix regression. Journal of the Royal Statistical Society Series B: Statistical Methodology, 76(2):463–483, 2014.

Hua Zhou, Lexin Li, and Hongtu Zhu. Tensor regression with applications in neuroimaging data analysis. Journal of the American Statistical Association, 108(502):540–552, 2013.

Shasha Zou and Aaron J Ridley. Modeling of the evolution of storm-enhanced density plume during the 24 to 25 October 2011 geomagnetic storm. Magnetosphere-Ionosphere Coupling in the Solar System, 10:205–213, 2016.

Shasha Zou, Mark B Moldwin, Michael J Nicolls, Aaron J Ridley, Anthea J Coster, Endawoke Yizengaw, Larry R Lyons, and Eric F Donovan. Electrodynamics of the high-latitude trough: Its relationship with convection flows and field-aligned currents. Journal of Geophysical Research: Space Physics, 118(5):2565–2572, 2013a.

Shasha Zou, Aaron J. Ridley, Mark B. Moldwin, Michael J. Nicolls, Anthea J. Coster, Evan G. Thomas, and J. Michael Ruohoniemi. Multi-instrument observations of SED during 24-25 October 2011 storm: Implications for SED formation processes. Journal of Geophysical Research: Space Physics, 118(12):7798–7809, December 2013b.

Shasha Zou, Mark B. Moldwin, Aaron J. Ridley, Michael J. Nicolls, Anthea J. Coster, Evan G. Thomas, and J. Michael Ruohoniemi. On the generation/decay of the storm-enhanced density plumes: Role of the convection flow and field-aligned ion flow. Journal of Geophysical Research: Space Physics, 119(10):8543–8559, October 2014.

Shasha Zou, Jiaen Ren, Zihan Wang, Hu Sun, and Yang Chen. Impact of storm-enhanced density (SED) on ion upflow fluxes during geomagnetic storm. Frontiers in Astronomy and Space Sciences, 8:162, 2021.