Advancing Climate Science with Machine Learning

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Sijie He

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Prof. Arindam Banerjee, Advisor

March, 2022

Acknowledgements

Foremost, I would like to express my deepest gratitude to my advisor Prof. Arindam Banerjee for his guidance, support, and encouragement throughout my Ph.D. study and research. I am sincerely grateful to him for giving me the chance to work on machine learning for advancing climate science, one of the most remarkable problems in my mind since I was a child. His enthusiasm for learning along with his intelligence and rigor in research have been inspiring to me all the time. I admire his patience and care with students, as well as his clarity of thought and precise communication. He can always "magically" explain complicated concepts in machine learning in a way that is easy to understand. What I have learned from him will benefit me endlessly.

I would like to thank Prof. Vipin Kumar, Prof. Timothy DelSole, and Prof. Daniel Boley for agreeing to be in my committee. All of them have generously provided help during my Ph.D. study and have raised insightful questions in my defense. I am honored to have such an incredible committee. In the past two and half years, I have worked closely with Prof. Timothy DelSole. He is one of the most knowledgeable and humble scientists I have ever met. His constructive guidance and support did have a significantly positive impact on my research. Prof. DelSole is a role model to me for being a responsible and respectful researcher. I am also grateful to my terrific collaborators Prof. Pradeep Ravikumar, Prof. Laurie Trenary, and Prof. Benjamin Cash - for their wonderful suggestions and genuine encouragement along the way.

My sincere thanks go to my labmates plus friends at the University of Minnesota, Twin Cities: Xinyan, Vidyashankar, Yingxue, Robert, Qilong, Konstantina, Sheng, Tiancong, Amir, and Farideh. I will miss the fun and educational group meetings and workshops we have had in Keller Hall. A special shoutout to Xinyan, Vidyashankar, and Yingxue, thank you for all the help and emotional support. I will never forget the late nights when we worked so hard together before those major deadlines. I am so happy and proud that we have all made it and become Ph.D. A big thank you to the awesome friends whom I met

in Minneapolis since the beginning of my Ph.D. - Xiaoxiao, Zeren, Haley, and Mengdie - for celebrating all the small victories with me and selflessly supporting me during all the difficult times over the past few years. I would love to take this opportunity to thank my friends in China - Ziling, Yushan, Haiyue, Qihui, and Xuejun. Even though we are not in the same country, your care and love have so many times warmed my heart and helped me overcome homesickness in the U.S. I also want to thank my friends - Zecheng, Yiyuan, and Jianyi - for having a lot of fun chats during the pandemic time.

My special appreciation goes to my family for their unconditional love and support. To my mother Yan, thank you for your love and care and for always believing in me! You are not only the most loving mom but also my best friend. You have been a role model for being an independent woman. It would not be possible for me to finish my Ph.D. without your continuous understanding and emotional support. To my father He, thank you for teaching me how to be a responsible and kind-hearted person through actions. I know no matter what happens, you will always be there for me. To my grandfather, thank you for instilling in me the interest in mathematics and encouraging me to pursue higher education since I was a little girl. To my best friend Bhaskar, thank you for rejoicing in each of my small achievements more than I did, for believing in me all the time, and for accompanying me through the challenging times.

Dedication

To my grandfather Yan Liang in loving memory.

ABSTRACT

Climate change is considered one of the greatest challenges for humanity in the twenty-first century. The changing climate affects almost every aspect of people's lives, including but not limited to water, energy, agriculture, ecosystems, economics, safety, and health. In the past decades, due to climate change, extreme events, such as wildfire, droughts, and flooding, have become more frequent and intensive, which can cause devastating economic loss and humanitarian crises. Therefore, skillful climate modeling, which can improve the understanding and predictability of climate behavior, would have immense societal values. In climate science, climate models are used for representing the major climate system components (atmosphere, land, ocean, and sea ice) and their interactions. A climate model consists of mathematical equations derived using fundamental laws of physics, which need to be solved using powerful supercomputers. In general, climate models are an important tool for understanding climate change, and continually become more complete and accurate. Nevertheless, the Earth's climate system is too complex to be fully simulated. The state-of-the-art climate models are not yet perfect for fulfilling all needs in understanding and forecasting climate behaviors, which leaves open opportunities for interdisciplinary climate studies.

In the past decades, machine learning (ML), especially deep learning, has achieved remarkable strides in wide-ranging applications. The emergence of climate data with high spatiotemporal resolution also makes it possible to tackle complex climate problems using machine learning techniques. Recent studies have shown the effectiveness of machine learning approaches on various tasks, including weather prediction, climate forecasting, weather extremes detection, etc. The dissertation explores how machine learning techniques can make advances in solving two fundamental problems in climate science. The first type of problem is on understanding the dependencies among or within key components in the climate system. Two types of machine learning models are proposed for addressing the problem, that are high-dimensional structure learning model and regularized regression model. We first propose a novel high-dimensional structure learning algorithm for estimating the underlying dependency structure (interactions) among different spatial locations around the globe in global atmospheric circulation. Secondly, to obtain a better understanding of the predictive relationships between land and ocean climate variables, we introduce a weighted Lasso model for land temperature prediction using sea surface temperatures and establish the finite sample estimation error bounds for the proposed model. The second climate problem that we target is sub-seasonal forecasting (SSF), the prediction of key climate variables, e.g., temperature and precipitation, on a 2-week to 2-month horizon. We investigate 10 machine learning models on SSF over the contiguous United States (U.S.). The experimental results indicate that suitable ML models can outperform commonly-used climate baselines and, to some extent, capture the predictability on sub-seasonal time scales. In addition, we perform a fine-grained comparison of a suite of modern ML models with state-of-the-art physics-based dynamical models for SSF over the western U.S. We carefully analyze the strengths of both types of models and propose to incorporate dynamical model forecasts in machine learning modeling, which significantly enhances the forecasting performance of the ML models. Further, to compensate for the limited availability of climate data for SSF, we work on generating synthetic climate data and propose a novel Vision Transformer-based variational autoencoder (ViT-VAE) model. We compare the proposed model with another dominant type of generative model, and show both models are able to generate realistic synthetic samples that match the underlying ground truth distribution closely.

Contents

List of Tables

List of Figures

Part I
Introduction

Chapter 1

Introduction

1.1 Background

Climate change is considered one of the greatest challenges for humanity in the twenty-first century (Romm, 2018). In addition to temperatures rising, climate change includes glaciers melting, sea level rise, more frequent weather extremes (e.g., drought and flooding), and much more. The changing climate affects almost every aspect of people's lives, including but not limited to water, energy, agriculture, ecosystems, economics, safety, and health. For example, the heatwave, which affected Western North America in June and July 2021, has led to a death toll of over 1,400 people and at least $8.9 billion in damages (NOAA, 2021). The heatwave has caused wildfires, snow melt, damages to crops, closures of businesses, destruction of road and rail infrastructure, etc. Such effects were seen as likely to trigger some more severe consequences, such as flooding and global food price increase.

Therefore, skillful climate modeling, which can improve our understanding and predictability of climate behavior, would have immense societal values. In climate science, climate models are used for representing the major climate system components (atmosphere, land, ocean, and sea ice) and their interactions (Edwards, 2011; Flato, 2011). The schematic of a climate model is shown in Figure 1.1. Specifically, the Earth's surface is divided into a three-dimensional grid of cells where the materials in each cell and the exchange of matter and energy with its neighbor cells can be described by mathematical equations. A climate model consists of systems of differential equations based on basic laws of physics, which need to be solved by powerful supercomputers (Edwards, 2011). Running and maintaining a climate model can be computationally expensive and labor-intensive, and sometimes require subjective decisions from climate experts (Hourdin et al., 2017). In general, global climate

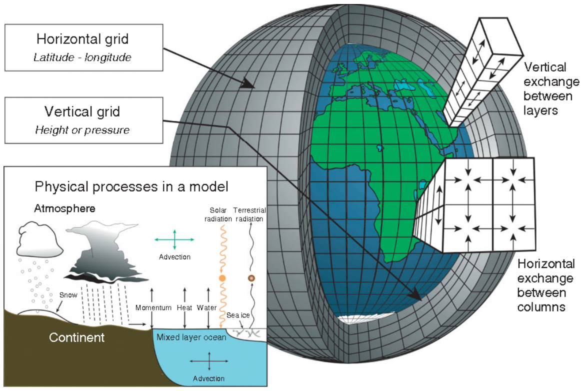


Figure 1.1: Schematic of a climate model coupling atmosphere, ocean, land, and sea ice. Graphic by Courtney Ritz and Trevor Burnham (Edwards, 2011).

models are useful for understanding the impact of human actions on climate change, while regional climate models are more practical for studying climate behavior for agriculture, local ecosystem, transportation system, etc. However, due to the high complexity of various climate processes and the limitation of computational resources, it is extremely hard to accurately model a climate system. Current climate models are not yet perfect to fulfill all the needs in understanding the relationships among various components in the climate systems as well as forecasting at different time scales, which leaves open opportunities for interdisciplinary climate studies.

In the past decades, machine learning, especially deep learning, has achieved remarkable strides in a wide range of applications, such as natural language processing, computer vision, speech recognition, etc (Devlin et al., 2018; Jordan and Mitchell, 2015; Krizhevsky et al., 2012a; LeCun et al., 2015; Vaswani et al., 2017). Meanwhile, as climate data with improved spatiotemporal resolutions have become available, increasing efforts have been made to tackle complex problems in climate science using machine learning models (Rolnick et al., 2019). For example, skillful machine learning methods, ranging from support vector machine to deep learning models, have been developed for short-term weather forecasting (Arcomano et al., 2020; Cofıno et al., 2002; Grover et al., 2015; Kuligowski and Barros, 1998; Radhika and Shashi, 2009; Ravuri et al., 2021), as well as for long-term climate forecasts (Badr et al., 2014; Cohen et al., 2019; Strobach and Bel, 2016). In addition, tailored machine learning solutions have been created for finding predictive relationships among different climate variables (Chatterjee et al., 2012; DelSole and Banerjee, 2017; He et al., 2019; Steinhaeuser et al., 2011b), and predicting weather extremes (Liu et al., 2016; McGovern et al., 2014; Racah et al., 2017). Such models have the potential to aid a better understanding of the impact of climate change and attribution of observed events as well as guide decision/policy making in a variety of domains such as agricultural planning, water resource management, and extreme weather events (O'Brien et al., 2006). In this dissertation, we develop machine learning models for two important problems in climate science, which are (1) understanding the dependencies among or within key components in the climate systems, and (2) forecasting climate variables on sub-seasonal time scales. We elaborate on the developments in machine learning for solving the two problems and illustrate our contributions in the subsequent section.

The first type of problem that we focus on is identifying the relationships within or among land, ocean, and atmosphere, which can help improve climate modeling and forecasting capabilities. There are two broad groups of machine learning methods that can be used for the task. The first group of methods is based on structure learning, which estimates a graph representing the dependence structure among different variables in a given dataset. In the past decades, structure learning has been emerging as an important tool for inferring interactions between spatiotemporal climate variables around the globe (Golmohammadi et al., 2017). Recent applications include the study of teleconnections (Chu et al., 2005) and global atmospheric information flow (de Perez and Mason, 2014; Ebert-Uphoff and Deng, 2012). In Chapter 2, we propose a novel structure learning algorithm to identify statistical dependencies in high-dimensional physical processes with a small sample size, which is shown capable of recovering the underlying structure of global atmospheric circulation. The second group of methods uses predictive modeling to identify the predictive relationships among various climate variables. A series of works have been conducted to predict land temperatures and precipitations from sea surface temperatures (SST), using machine learning techniques including principal component regression (PCR) (Francis and Renwick, 1998), clustering (Steinhaeuser et al., 2011a), neural networks (Hsieh and Tang, 1998), and high-dimensional linear regression (Chatterjee et al., 2012; DelSole and Banerjee, 2017). To obtain a better understanding of the predictive relationships between land and ocean climate variables, in Chapter 3, we introduce a weighted Lasso model for land temperature prediction using SST, which is capable of yielding interpretable results with high prediction accuracy.

As for the second problem, we shift focus towards a specific forecasting task in climate science, that is sub-seasonal climate forecasting (SSF). Sub-seasonal climate forecasting is the prediction of key climate variables, e.g., temperature and precipitation, on a 2-week to 2-month time horizon. Skillful sub-seasonal climate forecasts have significantly beneficial impacts on agricultural productivity, hydrology and water resource management, transportation and aviation systems, emergency planning for extreme weather events (National Academies of Sciences, Engineering, and Medicine, 2016; National Research Council, 2010). Despite its societal importance, the progress on SSF has remained limited (Braman et al., 2013; de Perez and Mason, 2014). In climate science, high-quality sub-seasonal forecasting has proven difficult to accomplish compared to both short-term weather forecasting and long-term seasonal forecasting (Vitart et al., 2012). Similarly, from a machine learning perspective, SSF poses an unconventional forecasting problem due to the unique nature of climate data and the large temporal gap (2 weeks to 2 months) between the latest available data and its forecasting target (He et al., 2021a). Recent studies (Buchmann and DelSole, 2021; He et al., 2021a; Hwang et al., 2019; Mouatadid et al., 2021; Srinivasan et al., 2021; Wang et al., 2021; Weyn et al., 2021) have shown that machine learning has the potential to advance sub-seasonal forecasting capabilities. In Chapter 4, we rigorously investigate 10 machine learning approaches to sub-seasonal temperature forecasting over the contiguous United States (U.S.), which shows suitable machine learning models can capture predictability at sub-seasonal time scales and outperform existing approaches in climate science. In Chapter 5, we perform a fine-grained comparison of a suite of modern machine learning models with state-of-the-art physics-based dynamical models from the Subseasonal Experiment (SubX) (Pegion et al., 2019) project for SSF in the western U.S. (He et al., 2021b). We carefully analyze the strengths of both types of models and propose to incorporate dynamical model forecasts into machine learning modeling, which significantly enhances the forecasting performance on sub-seasonal time scales. Further, the limited availability of high-quality climate data makes it more challenging for machine learning to advance SSF. Therefore, in Chapter 6, we focus on generating synthetic climate data using generative modeling and show that, with proper adjustment, deep generative models are able to generate synthetic data that match the ground truth distribution closely.

1.2 Contributions

In this dissertation, we dive deep into two fundamental climate problems and develop corresponding machine learning solutions. In Part II, including Chapter 2 and 3, we target on identifying relationships among or within land, ocean, and atmosphere. Part III, including Chapter 4, Chapter 5, and Chapter 6, focuses on advancing sub-seasonal climate forecasting using machine learning. Below we introduce the contribution of the subsequent chapters in detail and clarify the specific problems solved in each chapter.

In Chapter 2, we consider the use of structure learning methods for probabilistic graphical models to identify statistical dependencies in high-dimensional physical processes (Golmohammadi et al., 2017). Such processes are often synthetically characterized using partial differential equations (PDEs) and are observed in a variety of natural phenomena. We present ACLIME-ADMM, an efficient two-step algorithm for adaptive structure learning, which decides a suitable edge-specific threshold in a data-driven statistically rigorous manner. We compare ACLIME-ADMM with classical (like PC algorithm (Spirtes et al., 2000)) and modern (like CLIME algorithm (Cai et al., 2011)) structure learning approaches on both synthetic data that model advection-diffusion processes, and real data (50 years) of daily global geopotential heights. ACLIME-ADMM is shown to be efficient, stable, and competitive, especially can outperform the baselines in difficult scenarios. On real data, ACLIME-ADMM recovers the underlying structure of global atmospheric circulation, including switches in wind directions at the equator and tropics entirely from the data.

In Chapter 3, we consider the problem of predicting monthly deseasonalized land temperature at different locations worldwide using sea surface temperature (He et al., 2019). Contrary to popular belief on the trade-off between (a) simple interpretable but inaccurate models and (b) complex accurate but uninterpretable models, we introduce a weighted Lasso model for the problem which yields interpretable results while being highly accurate. In addition, we establish finite sample estimation error bounds for weighted Lasso, and illustrate its superior empirical performance and interpretability over complex models, such as deep neural networks (Deep nets) (Goodfellow et al., 2016) and gradient boosted trees (GBT) (Chen and Guestrin, 2016; Friedman, 2001) . We also present a detailed empirical analysis of what has been wrong with Deep nets for the specific problem, which may serve as a helpful guideline for the application of Deep nets to small sample scientific problems.

In Chapter 4, we carefully investigate 10 machine learning approaches to sub-seasonal temperature forecasting over the contiguous U.S. using the SSF dataset we collect (He et al., 2021a). The SSF dataset includes a variety of climate variables and climate indices from the atmosphere, ocean, and land. Our results indicate that suitable machine learning models, e.g., XGBoost (Chen and Guestrin, 2016), to some extent, capture the predictability on sub-seasonal time scales and can outperform the climatological baselines, while deep learning models barely manage to match the best results with carefully designed architectures. Besides, our analysis and exploration provide insights on important aspects to improve the quality of sub-seasonal forecasts, e.g., feature representation and model architecture. The SSF dataset and code base have been made publicly available for the broader research community.

In Chapter 5, we perform a fine-grained comparison of a suite of machine learning models with state-of-the-art physics-based dynamical models from the SubX project for SSF over the western U.S. (He et al., 2021c). Empirical results illustrate that, on average, machine learning models outperform dynamical models, while the machine learning models are more likely to underpredict the amplitude compared to the SubX models. Further, we show that machine learning models make forecasting errors under extreme weather conditions, e.g., cold waves due to the polar vortex, highlighting the need for separate models for extreme events. Finally, we explore potential mechanisms to enhance machine learning models and show that suitably incorporating dynamical model forecasts as inputs to machine learning models can substantially improve the forecasting performance of the machine learning models.

In Chapter 6, to compensate for the limited availability of high-quality climate data for SSF, we seek to generate synthetic 2-meter temperature (tmp2m) anomalies using deep generative models. We propose a novel ViT-based variational autoencoder (ViT-VAE) model which combines the state-of-the-art computer vision model with VAE. The proposed model can learn latent representations of tmp2m anomalies for each climatically consistent region, and generate synthetic climate data over the western U.S. In addition, we carefully compare the proposed model with another popular type of generative model, i.e., Wasserstein Generative Adversarial Networks with Gradient Penalty (WGAN-GP). The empirical results illustrate that, with proper adjustment, deep generative models are able to generate synthetic tmp2m anomalies that match the ground truth distribution closely.

Part II
Identify Dependencies Among Key Climate Variables

Chapter 2

Learning Statistical Dependencies in High-Dimensional Physical Processes

2.1 Introduction

The ability to infer interactions between variables from high-dimensional data sets has the potential to help geoscientists answer numerous questions critical for improved modeling and prediction capabilities for various geoscience processes. Using atmospheric science as an example, it would enable us to (1) delineate better the interactions between atmospheric disturbances of different spatial scales, which is critical for understanding the working of a weather-climate continuum; (2) develop a better understanding of the degree and spatial pattern of coupling between the top of atmosphere (TOA) radiative imbalance and surface temperatures, which provides a unique perspective of climate feedback processes; (3) identify causal pathways in the atmospheric circulation and infer how they might change under a warming climate (Deng and Ebert-Uphoff, 2014); and (4) study the dynamical processes of air-sea interaction that lead to the onset of the monsoons. These applications would contribute to both our understanding of the key processes determining the main features of the Earth's climate system and our capabilities to predict changes in this system with changing external forcing (e.g., aerosols and greenhouse gas emissions) in the near future.

Structure learning is thus emerging in the geoscience as an important tool for that purpose. Recent applications include the study of tele-connections (Chu et al., 2005) and the study of atmospheric information flow around the globe (Ebert-Uphoff and Deng, 2012).

Such studies have only recently become possible, thanks to increasing computational power, combined with the rapidly increasing amount of observational and model output data for the earth atmosphere (Stocker et al., 2013).

2.1.1 State-of-the-art and Its Limitations

Structure learning methods can be broadly divided into two groups. The first group of methods were developed in the seminal work by Pearl (Pearl, 2009) and Spirtes-Glymour-Scheines (Spirtes et al., 2000), among others. The PC algorithm and its variants (Spirtes et al., 2000), (Spirtes and Glymour, 1991), (Kalisch and Bühlmann, 2007), (Harris and Drton, 2013) constitute the most popular methods from this family, and are capable of producing the skeletal structure of the underlying Bayesian network capturing the data dependency. However, such methods are 'information-theoretic' in the sense that they give the correct output in the asymptotic limit of infinite samples (Kalisch and Bühlmann, 2007) and may need exponential computation in the worst case. On the statistical side, in the real world setting of finite samples, such methods cannot (yet) characterize the probability of error (or p-value) of the graph produced. On the computational side, while advances have been made (Colombo and Maathuis, 2014), existing advanced implementations of the PC algorithm do not scale beyond 100,000 variables, whereas geoscience data routinely involves higher dimensional physical processes (Karpatne et al., 2017).

The second group of methods, such as graphical Lasso (Friedman et al., 2008), (Meinshausen and Bühlmann, 2006) and CLIME (Cai et al., 2011), have seen active development over the past decade (Cai et al., 2016), (Wang et al., 2013) and come with rigorous finite sample statistical guarantees and efficient computational algorithms. However, such algorithms do need to assume the joint distribution over the variables to be of a specific (semi)parametric family, e.g., multivariate Gaussian (copula), Ising, multivariate Poisson, etc. The second group of methods (Drton and Maathuis, 2017), based on sparse high-dimensional estimation, can do structure learning by estimating the moral graph of the underlying Bayes net using finite samples *in theory*, but has a major limitation *in practice*: *instability* due to (hyper-)parameter choices. Such methods, based on Lasso and variants need to choose constants, say λ for Lasso (Friedman et al., 2008), (Meinshausen and Bühlmann, 2006), which determine the level of sparsity. For structure learning, the output graph can vary significantly based on the specific parameters used. Recent years have seen advances on making the output more stable possibly by repeatedly running the algorithm for different values of the parameters possibly on (disjoint) subsets of the sample (Meinshausen and Bühlmann, 2010), (Meinshausen and Bühlmann, 2006). Such advances, while promising, are computationally demanding, due to the need for repeated runs, and can be statistically demanding due to the need for larger samples.

2.1.2 Contributions of This Work

We seek to address the issues of both stability and computational demands in this work through the following key contributions. First, we introduce ACLIME-ADMM, an efficient two-step algorithm for adaptive structure learning, which estimates an edge specific parameter λ_{ij} for edge (i, j) in the first step, and uses these parameters to learn the structure in the second step. Both steps of our algorithm use (inexact) ADMM to solve suitable linear programs, and all iterations can be done in closed form. Second, we propose a significantly more scalable version of ACLIME-ADMM based on block updates rather than in single column updates for basic ACLIME-ADMM. The block updates are non-trivial since every column solves a mildly different linear program. The proposed method is developed based on a careful analysis of the shared structure of these problems, and first does a block update followed by column specific adjustments. Third, we illustrate the effectiveness of ACLIME-ADMM by comparisons with state-of-the-art baselines, i.e, PC-variants (PC stable (Colombo and Maathuis, 2014)) and CLIME variants (CLIME-ADMM (Wang et al., 2013)) through extensive experiments on both synthetic and real data involving geo-physical processes. Furthermore, methods from *structure learning* for probabilistic graphical models (Drton and Maathuis, 2017; Pearl, 1988) have been applied with great success in disciplines ranging from social science (Spirtes et al., 2000) to bioinformatics (Chen et al., 2006), to identify *direct* dependencies. The proposed algorithm can also be applied in such area with its advantages of efficiency and scaliablity.

The rest of the chapter is organized as follows. We elaborate our derivation of ACLIME-ADMM algorithm in Section 2.2, along with the stability analysis for hyperparameters. In section 2.3, PC stable algorithm and how structure learning algorithm is applied for temporal models are illustrated. We provide the description of both synthetic and observed data sets for climate application and the corresponding experimental results in section 2.4 and section 2.5 respectively. The advantages of fast implementation of the proposed algorithm is illuminated in section 2.6 and the chapter is concluded in section 2.7.

2.2 Related Work

Over the past decade, advances in structure learning have been made by making explicit assumptions about the parametric form of the joint distribution. For example, advances have been made based on the assumption that the joint distribution is a multivariate Gaussian (Cai et al., 2011; Friedman et al., 2008; Meinshausen and Bühlmann, 2006), or a Gaussian copula distribution (Liu et al., 2012; Xue et al., 2012). Typically, such estimators involve a sparsity inducing optimization problem, and efficient algorithms for solving such problems have been developed (Banerjee et al., 2008; Hsieh et al., 2011). In recent work, the CLIME estimator (Cai et al., 2011) was proposed to estimate sparse inverse of covariance matrix (precision matrix), which reveals the dependency structure for multivariate Gaussian distribution (Lauritzen, 1996). For a p-dimensional problem, CLIME estimates the sparse precision matrix $\Omega \in \mathbb{R}^{p \times p}$ by solving the following linear program (LP):

$$\hat{\Omega} = \underset{\Omega \in \mathbb{R}^{p \times p}}{\operatorname{argmin}} \|\Omega\|_1 \quad \text{s.t.} \quad \|C\Omega - I\|_\infty \leq \lambda \ , \tag{2.1}$$

where $C \in \mathbb{R}^{p \times p}$ is the covariance matrix and $\lambda > 0$ is a tuning parameter. Recent work has developed scalable optimization algorithms for the problem, which have been shown to scale to a million dimensions (Wang et al., 2013). In spite of its scalability, the empirical performance of the CLIME estimator is sensitive to the choice of the tuning parameter λ, and it is usually difficult to make the choice in a rigorous data driven manner (Cai et al., 2011; Wang et al., 2013). In recent work, a more powerful adaptive version of CLIME, called ACLIME, has been proposed (Cai et al., 2016). In this section, we propose the ACLIME-ADMM algorithm, which is able to solve the corresponding optimization efficiently using block parallel updates along with simple per column adjustments. The introduced inexact ADMM algorithm, which utilizes closed-form updates for both primal and dual variables, improves the scalability of our method considerably.

2.3 Adaptive Estimation of Statistical Dependencies

While estimators such as graphical Lasso (Friedman et al., 2008; Meinshausen and Bühlmann, 2006) and CLIME (Cai et al., 2011) effectively use the same (soft/box) threshold parameter λ, recent work on the Adaptive CLIME (Cai et al., 2016) estimator advocates using a different threshold parameter λ_{ij} for different entries. Such a choice arguably leads to better statistical properties of the estimator (Cai et al., 2016). Further, the necessary threshold parameters themselves can be obtained in a data driven manner using a suitable estimator.

2.3.1 *ACLIME* Estimator

We start by briefly reviewing the ACLIME estimator, the key optimization problems which need to be solved. The following result (Cai et al., 2016) motivates the estimator:

Theorem 1. *Let $x_1, \cdots, x_n \sim N_p(\mu^*, C^*)$ with $\log p = O(n^{1/3})$, and let Ω^* be the corresponding precision matrix. Let C be the unbiased sample estimate of C^* and let $S = (s_{ij})_{1 \leq i,j \leq p} = C\Omega^* - \mathbb{I}_{p \times p}$. Then*

$$Var(s_{ij}) = \begin{cases} n^{-1}(1 + c^*_{ii}\omega^*_{ii}) & for \quad i = j \\ \quad n^{-1}c^*_{ii}\omega^*_{jj} & for \quad i \neq j \ , \end{cases}$$

and for all $\delta \geq 2$,

$$\mathbb{P}\left\{ |(C\Omega^* - \mathbb{I}_{p\times p})_{i,j}| \leq \delta\sqrt{\frac{c^*_{ii}\omega^*_{jj}\log p}{n}}, \forall 1 \leq i, j \leq p \right\} \geq 1 - O\left((\log p)^{-\frac{1}{2}} p^{-\frac{\delta^2}{4}+1} \right) \ . \quad (2.2)$$

To use the adaptive bound in (2.2), one can use the sample estimate c_{ii} as a surrogate to c^*_{ii}. However, the bound also needs an estimate of ω^*_{jj}, the diagonal estimates of the precision matrix. The ACLIME estimator works in two stages: in the first stage, an estimate $\breve{\omega}_{jj}$ for ω^*_{jj} is computed; in the second stage, the estimate $\breve{\omega}_{jj}$ is used to adaptively estimate Ω based on (2.2). In particular, *in the first stage*, each column of the precision matrix is estimated (Cai et al., 2016) by solving:

$$\hat{\omega}^1_{\cdot j} = \underset{\mathbf{b}_j \in \mathbb{R}^p}{\text{argmin}} \{ \|\mathbf{b}_j\|_1 : |\hat{C}\mathbf{b}_j - \mathbf{e}_j|_\infty \leq \tau_n(c_{ii} \vee c_{jj}) \times b_{jj}, b_{jj} > 0 \} \ , \quad (2.3)$$

where $\hat{C} = C + \frac{1}{n}\mathbb{I}_{p\times p}$, $\tau_n = \delta\sqrt{\frac{\log p}{n}}$, $\delta \geq 2$, $(c_{ii} \vee c_{jj}) = \max(c_{ii}, c_{jj})$ and b_{jj} is the j-th element in $\mathbf{b}_j$. Then, the diagonal elements ω^*_{jj} are estimated as:

$$\breve{\omega}_{jj} = \hat{\omega}^1_{jj} I \left\{ c_{jj} \leq \sqrt{\frac{n}{\log p}} \right\} + \sqrt{\frac{\log p}{n}} I \left\{ c_{jj} > \sqrt{\frac{n}{\log p}} \right\} \ . \quad (2.4)$$

Given $\breve{\omega}_{jj}$, *in the second stage*, ACLIME estimates Ω^* by first solving the following optimization problem to get a primitive estimate of the j-th column:

$$\tilde{\omega}^1_{\cdot j} = \underset{\mathbf{b}_j \in \mathbb{R}^p}{\text{argmin}} \{ \|\mathbf{b}_j\|_1 : |(\hat{C}\mathbf{b}_j - \mathbf{e}_j)_i| \leq \tau_n\sqrt{c_{ii}\breve{\omega}_{jj}} \} \ . \quad (2.5)$$

In the final step, ACLIME symmetrizes $\tilde{\Omega}^1 = (\tilde{\omega}^1_{ij})$ to obtain $\hat{\Omega} = (\hat{\omega}_{ij})$, the estimate of Ω^*:

$$\hat{\omega}_{ij} = \hat{\omega}_{ji} = \tilde{\omega}^1_{ij} I\{|\tilde{\omega}^1_{ij}| \leq |\tilde{\omega}^1_{ji}|\} + \tilde{\omega}^1_{ji} I\{|\tilde{\omega}^1_{ij}| > |\tilde{\omega}^1_{ji}|\} \ . \tag{2.6}$$

2.3.2 *ACLIME-ADMM* Algorithm

We now focus on developing efficient optimization algorithms for solving the two stages of the ACLIME estimation, in particular the problems in (2.3) and (2.5). (Cai et al., 2016) observes that the optimization problem can be decomposed into p independent LPs, one for each column of $\hat{\Omega}$. We first introduce an inexact ADMM algorithm for solving the column-specific LPs corresponding to each stage, where all computations are in closed form based on elementwise operations and matrix multiplications. Later we generalize the algorithm to solve column block LPs where the computations need more care since the LP for each column is mildly different but has some shared structure which our algorithm uses. As the experiments illustrate, the methods are efficient and scalable.

Stage 1: Estimating diagonal elements ω_{jj}. We first focus on developing an approach to solving (2.3), which yields the initial estimates of the diagonal elements ω_{jj} of the precision matrix. We z-score the variables so that $c_{jj} = 1$ for $j = 1, \ldots, p$. As a result, considering the constraint in (2.3), we note that $\tau_n(c_{ii} \vee c_{jj}) = \tau_n$. Hence the constraint in (2.3) can be rewritten as:

$$-\tau_n b_{jj} \mathbf{1}_p \leq \hat{C}\mathbf{b}_j - \mathbf{e}_j \leq \tau_n b_{jj} \mathbf{1}_p \ , \tag{2.7}$$

where $\mathbf{1}_p$ is the p dimensional vector with all entries being 1. Focusing on the right hand side inequality in (2.7), we can rewrite it as:

$$\hat{C}_{\text{up}} \mathbf{b}_j \leq \mathbf{e}_j \ , \tag{2.8}$$

where $\hat{C}_{\text{up}} = \hat{C} - \tau_n \mathbf{1}_p \mathbf{e}_j^T$. Note that $\hat{C}_{\text{up}}$ is a rank-1 and sparse perturbation of $\hat{C}$ where only column j, interacting with b_{jj}, gets a constant τ_n subtracted from every entry. Introducing non-negative variables $\mathbf{u}_j \in \mathbb{R}^p_+$, so that $\mathbf{u}_j \geq \mathbf{0}_p$, the p dimensional vector with all entries being 0, the inequality constraint in (2.8) can be rewritten as an equality constraint:

$$\hat{C}_{\text{up}} \mathbf{b}_j + \mathbf{u}_j = \mathbf{e}_j \ . \tag{2.9}$$

Similarly, focusing on the left hand side inequality of (2.7), we get

$$-\hat{C}_{\text{down}}\mathbf{b}_j \leq -\mathbf{e}_j \tag{2.10}$$

where $\hat{C}_{\text{down}} = \hat{C} + \tau_n \mathbf{1}_p \mathbf{e}_j^T$. Introducing non-negative variables $\mathbf{v}_j \in \mathbb{R}_+^p$, so that $\mathbf{v}_j \geq \mathbf{0}_p$, the inequality constraint in (2.10) can be rewritten as an equality constraint:

$$-\hat{C}_{\text{down}}\mathbf{b}_j + \mathbf{v}_j = -\mathbf{e}_j \ , \tag{2.11}$$

Then, by combining (2.9) and (2.11), the constraint corresponding to (2.7) can be written as:

$$\underbrace{\begin{bmatrix} \hat{C}_{\text{up}} \\ -\hat{C}_{\text{down}} \end{bmatrix}}_{A_j} \mathbf{b}_j + \underbrace{\begin{bmatrix} \mathbb{I}_{p\times p} & 0 \\ 0 & \mathbb{I}_{p\times p} \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} \mathbf{u}_j \\ \mathbf{v}_j \end{bmatrix}}_{\mathbf{r}_j} = \underbrace{\begin{bmatrix} \mathbf{e}_j \\ -\mathbf{e}_j \end{bmatrix}}_{\mathbf{c}_j} \ . \tag{2.12}$$

Then, the original problem in (2.3) can be written in a canonical form suitable for ADMM as follows:

$$\min_{\mathbf{b}_j\in\mathbb{R}^p, \mathbf{r}_j\in\mathbb{R}^{2p}} \ \|\mathbf{b}_j\|_1 + \mathbb{1}_{\mathbb{R}_+}(\mathbf{r}_j) \quad \text{s.t.} \quad A_j\mathbf{b}_j + \mathbf{r}_j = \mathbf{c}_j \ , \tag{2.13}$$

where $\mathbb{1}_{\mathbb{R}_+^{2p}}(\cdot)$ is the indictor function over non-negative reals in $\mathbb{R}^{2p}$, i.e., $\mathbb{1}_{\mathbb{R}_+^{2p}}(\mathbf{z}_j) = 0$, if $\mathbf{z}_j \geq \mathbf{0}_{2p}$, and ∞ otherwise, and we have used the fact $B = \mathbb{I}_{2p\times 2p}$, the identity matrix.

The augmented Lagrangian of the optimization problem in (2.13) is :

$$L(\mathbf{b}_j, \mathbf{r}_j, \mathbf{y}_j) = \|\mathbf{b}_j\|_1 + \mathbb{1}_{\mathbb{R}_+^{2p}}(\mathbf{r}_j) + \rho\langle \mathbf{y}_j, A_j\mathbf{b}_j + \mathbf{r}_j - \mathbf{c}_j\rangle + \frac{\rho}{2}\|A_j\mathbf{b}_j + \mathbf{r}_j - \mathbf{c}_j\|_2^2 \ , \tag{2.14}$$

where $\mathbf{y}_j \in \mathbb{R}^{2p}$ is the Lagrange multiplier vector. Based on the augmented Lagrangian, the ADMM steps are:

$$\mathbf{b}_j^{t+1} = \underset{\mathbf{b}_j\in\mathbb{R}^p}{\text{argmin}} \|\mathbf{b}_j\|_1 + \frac{\rho}{2}\|A_j\mathbf{b}_j + \mathbf{r}_j^t - \mathbf{c}_j + \mathbf{y}_j^t\|_2^2 \tag{2.15a}$$

$$\mathbf{r}_j^{t+1} = \underset{\mathbf{r}_j\in\mathbb{R}^{2p}}{\text{argmin}} \mathbb{1}_{\mathbb{R}_+^{2p}}(\mathbf{r}_j) + \frac{\rho}{2}\|A_j\mathbf{b}_j^{t+1} + \mathbf{r}_j - \mathbf{c}_j + \mathbf{y}_j^t\|_2^2 \tag{2.15b}$$

$$\mathbf{y}_j^{t+1} = \mathbf{y}_j^t + A_j\mathbf{b}_j^{t+1} + \mathbf{r}_j^{t+1} - \mathbf{c}_j \ . \tag{2.15c}$$

The update of $\mathbf{b}_j$ in (2.15a) does not have a closed form solution because the $A_j^T A_j$ term makes the components of $\mathbf{b}_j$ coupled. While one can use iterative approaches to solve the problem, we decouple the $\mathbf{b}_j$ by linearizing the quadratic term and adding a proximal term, a strategy used in inexact ADMM (Boyd et al., 2011):

$$\mathbf{b}_j^{t+1} = \arg\min_{\mathbf{b}_j \in \mathbb{R}^p} \|\mathbf{b}_j\|_1 + \eta \langle \mathbf{g}_j^t, \mathbf{b}_j \rangle + \frac{\eta}{2} \|\mathbf{b}_j - \mathbf{b}_j^t\|_2^2 \ , \tag{2.16}$$

where $\mathbf{g}_j^t = \frac{\rho}{\eta} A_j^T (A_j \mathbf{b}_j^t + \mathbf{r}_j^t - \mathbf{c}_j + \mathbf{y}_j^t)$ and $\eta > 0$. Inexact ADMM has been shown to have the same rate of convergence as ADMM for general (non-smooth) convex optimization problems (Wang and Banerjee, 2014). Now, based on the dual update in (2.15c), we have $\mathbf{g}_j^t = \frac{\rho}{\eta} A_j^T (2\mathbf{y}_j^t - \mathbf{y}_j^{t-1})$. Then, (2.16) has the following closed form solution based on soft-thresholding (Boyd et al., 2011)

$$\mathbf{b}_j^{t+1} = \text{soft}(\mathbf{b}_j^t - \mathbf{g}_j^t, \frac{1}{\eta}) \ . \tag{2.17}$$

Updating $\mathbf{r}_j^{t+1}$ in (2.15b) is simply the projection of elements of $\mathbf{h}_j^t = \mathbf{c}_j - \mathbf{y}_j^t - A_j \mathbf{b}_j^{t+1}$ to $\mathbb{R}_+^{2p}$ which can be done in closed form as $\mathbf{r}_j^{t+1} = \max(\mathbf{h}_j^t, 0)$, applied elementwise.

The solution of the above optimization for stage 1 gives $\hat{\omega}_{\cdot j}^1$ in (2.3), from which only the diagonal elements $\hat{\omega}_{jj}^1$ are of interest, which are then used to compute $\breve{\omega}_{jj}$ following (2.4).

Stage 2: Estimating Ω. In the second stage of ACLIME, the goal is to utilize the $\breve{\omega}_{jj}$ estimated in stage 1, and solve the problem in (2.5) to obtain $\tilde{\omega}_{\cdot j}$. Considering the constraints in (2.5), since $c_{ii} = 1$ due to z-scoring, the constraints over $\mathbf{b}_j \in \mathbb{R}^p$ can be simplified to

$$-\tau_n \sqrt{\breve{\omega}_{jj}} \mathbf{1}_p \leq \hat{C} \mathbf{b}_j - \mathbf{e}_j \leq \tau_n \sqrt{\breve{\omega}_{jj}} \tag{2.18}$$

Then, following the same strategy as used for stage 1, the system of linear inequality constraints can be rewritten as a system of equality constraints

$$\underbrace{\begin{bmatrix} \hat{C} \\ -\hat{C} \end{bmatrix}}_{A} \mathbf{b}_j + \underbrace{\begin{bmatrix} I_{p\times p} & 0 \\ 0 & I_{p\times p} \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} \mathbf{u}_j \\ \mathbf{v}_j \end{bmatrix}}_{\mathbf{r}_j} = \underbrace{\begin{bmatrix} \mathbf{e}_j + \tau_n \sqrt{\breve{\omega}_{jj}} \mathbf{1}_p \\ -\mathbf{e}_j + \tau_n \sqrt{\breve{\omega}_{jj}} \mathbf{1}_p \end{bmatrix}}_{\mathbf{c}_j} \ . \tag{2.19}$$

where $\mathbf{r}_j \in \mathbb{R}_+^{2p}$ as before. Then, the original problem in (2.5) can be written in a canonical form suitable for ADMM as follows:

$$\min_{\mathbf{b}_j \in \mathbb{R}^p, \mathbf{z}_j \in \mathbb{R}^{2p}} \|\mathbf{b}_j\|_1 + \mathbb{1}_{\mathbb{R}_+}(\mathbf{z}_j) \quad \text{s.t.} \quad A\mathbf{b}_j + \mathbf{z}_j = \mathbf{c}_j \ . \tag{2.20}$$

We note that the optimization problem in (2.13) is essentially the same as that in (2.20), in fact simpler since A is the same for all j. One can use the same ADMM algorithm for stage 2, take advantage of the same structures in the matrices to speed up computations, and also perform block updates which are going to be simpler since A is the same for all j.

Given that the structure of the optimization in stage 2 is simpler, one can also consider an alternative route (Wang et al., 2013), which uses less variables and is arguably amenable to block updates. Note that since $c_{ii} = 1$ due to z-scoring, the problem in (2.5) can be posed as:

$$\min_{\mathbf{b}_j \in \mathbb{R}^p} \ \|\mathbf{b}_j\|_1 \quad \text{s.t.} \quad \|\hat{C}\mathbf{b}_j - \mathbf{e}_j\|_\infty \leq \lambda_j \tag{2.21}$$

where $\lambda_j = \tau_n \sqrt{\breve{\omega}_{jj}}$ is a constant. Introducing $\mathbf{z}_j \in \mathbb{R}^p$, the problem can be rewritten as

$$\min_{\mathbf{b}_j, \mathbf{z}_j \in \mathbb{R}^p} \ \|\mathbf{b}_j\|_1 \quad \text{s.t.} \quad \|\mathbf{z}_j - \mathbf{e}_j\|_\infty \leq \lambda_j \ , \hat{C}\mathbf{b}_j = \mathbf{z}_j \ . \tag{2.22}$$

Note that the constraint on $\mathbf{z}_j$ is a box constraint, on which efficient projection is possible. Hence the box constraint can be handled inside the primal update for $\mathbf{z}_j$, without having to convert the box constraint to a system of equality constraints. Thus, ignoring the box constraint for now, the augmented Lagrangian is

$$L(\mathbf{b}_j, \mathbf{r}_j, \mathbf{y}_j) = \|\mathbf{b}_j\|_1 + \rho\langle \mathbf{y}_j, \hat{C}\mathbf{b}_j - \mathbf{z}_j\rangle + \frac{\rho}{2}\|\hat{C}\mathbf{b}_j - \mathbf{z}_j\|^2 \ . \tag{2.23}$$

The ADMM updates, which take the box constraint into account, are as follows

$$\mathbf{b}_j^{t+1} = \underset{\mathbf{b}_j \in \mathbb{R}^p}{\operatorname{argmin}} \|\mathbf{b}_j\|_1 + \frac{\rho}{2}\|\hat{C}\mathbf{b}_j - \mathbf{z}_j^t + \mathbf{y}_j^t\|^2 \tag{2.24a}$$

$$\mathbf{z}_j^{t+1} = \underset{\|\mathbf{z}_j - \mathbf{e}_j\|_\infty \leq \lambda_j}{\operatorname{argmin}} \frac{\rho}{2}\|\hat{C}\mathbf{b}_j^{t+1} - \mathbf{z}_j + \mathbf{y}_j^t\|^2 \tag{2.24b}$$

$$\mathbf{y}_j^{t+1} = \mathbf{y}_j^t + \hat{C}\mathbf{b}_j^{t+1} - \mathbf{z}_j^{t+1} \ . \tag{2.24c}$$

Note that (2.24a) can be solved using an inexact update similar to (2.16). Further, we note that the box-constrained quadratic problem in (2.24b) can be solved in closed form as

$$\mathbf{z}_j^{t+1} = \text{box}(\hat{C}\mathbf{b}_j^{t+1} + \mathbf{y}_j^t, \mathbf{e}_j, \lambda_j) \tag{2.25}$$

where for $\mathbf{a}, \mathbf{w} \in \mathbb{R}^p, \lambda \in \mathbb{R}_+$

$$\text{box}(\mathbf{a}, \mathbf{w}, \lambda) = \begin{cases} w_i + \lambda \ , & \text{if } \ a_i - w_i > \lambda \\ a_i \ , & \text{if } \ |a_i - w_i| \leq \lambda \\ w_i - \lambda \ , & \text{if } \ a_i - w_i < -\lambda \ . \end{cases} \tag{2.26}$$

In the current setting, $\mathbf{a} = \hat{C}\mathbf{b}_j^{t+1} + \mathbf{y}_j^t, \mathbf{w} = \mathbf{e}_j$, and $\lambda = \lambda_j$.

The solution of the above optimization in stage 2 gives $\tilde{\omega}_{\cdot j}$ in (2.5). The final step is to symmetrize the resulting precision matrix estimate as in (2.6).

2.3.3 Column-Block *ACLIME-ADMM* Algorithm

We propose an improvement to solve the two-stage ACLIME optimization in terms of column blocks instead of column-by-column.The implementation for each step is either element-wise parallel or utilizes suitable matrix multiplication, which improved the computational efficiency of the proposed algorithm. For stage one, we rewrite $A_j\mathbf{b}_j$ as following:

$$A_j\mathbf{b}_j = \begin{bmatrix} \hat{C}_{\text{up}}\mathbf{b}_j \\ \hat{C}_{\text{down}}\mathbf{b}_j \end{bmatrix} = \begin{bmatrix} \hat{C}x - \tau_n b_{jj}\mathbf{1}_p \\ -\hat{C}x - \tau_n b_{jj}\mathbf{1}_p \end{bmatrix} = \begin{bmatrix} \hat{C} \\ -\hat{C} \end{bmatrix}\mathbf{b}_j - \tau_n b_{jj}\mathbf{1}_{2p} \ . \tag{2.27}$$

Since all A_j are transformed from $\hat{C}$, the computation across columns can be shared, e.g., computing $\hat{C}\mathbf{b}_j$. Now we consider the column blocks, assuming $X \in \mathbb{R}^{p\times k}$ denotes k columns in $\hat{\Omega}$. Thus, the AX for a column block is defined as:

$$AX = \begin{bmatrix} \hat{C} \\ -\hat{C} \end{bmatrix} X - \mathbf{1}_{2p\times k}X_{\text{diag}} \ , \tag{2.28}$$

where $X_{\text{diag}} \in \mathbb{R}^{k\times k}$ is a diagonal matrix with the corresponding diagonal elements in X and $\mathbf{1}_{2p\times k} \in \mathbb{R}^{2p\times k}$ is a matrix with all entries being 1. Therefore, the equality constraints (2.12) for column block is $AX + R = E$, where R is the column block of corresponding $\mathbf{r}_j$ in (2.13) and $E \in \mathbb{R}^{p\times k}$ denotes the same k columns in $\mathbb{I}_{p\times p}$.

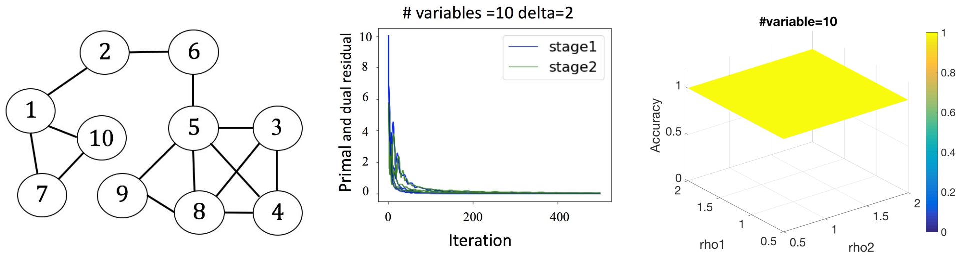
Therefore, the optimization problem is rewritten as follows:

$$\begin{aligned} \min_{X\in\mathbb{R}^{p\times k}, R\in\mathbb{R}^{2p\times k}} & \|X\|_1 + \mathbb{1}_{\mathbb{R}_+^{2p\times k}}(R) \\ \text{s.t.} \quad & AX + R = Z \ . \end{aligned} \tag{2.29}$$

Thus, the augmented Lagrangian of the above optimization problem is

$$L_\rho = \|X\|_1 + \mathbb{1}_{\mathbb{R}_+^{2p\times k}}(R) + \rho\langle Y, AX + R - E\rangle + \frac{\rho}{2}\|AX - R - E\|_2^2, \tag{2.30}$$

where $Y \in \mathbb{R}^{p\times k}$ is a scaled dual variable and $\rho > 0$. Similar to (2.24), inexact ADMM



(a) Undirected Graph (b) Residuals (c) Accuracy

Figure 2.1: The results of 10-variable synthetic data from column-block ACLIME-ADMM. (a) It is the true underlying undirected graph for the synthetic data, which illustrates the dependencies among 10 variables. (b) The primal and dual residuals of two stages in column-block ACLIME-ADMM are shown converging to 0 after 400 iterations when $\delta = 2$. (c) When parameter ρ in two stages are chosen in $(0, 2]$, the accuracy for estimating the non-zero entries in precision matrix are always 100% for the synthetic data.

yields the following iterates:

$$X^{t+1} = \underset{X \in \mathbb{R}^{p\times k}}{\operatorname{argmin}} \|X\|_1 + \eta\langle V^t, X\rangle + \frac{\eta}{2}\|X - X^t\|_2^2 \tag{2.31a}$$

$$R^{t+1} = \underset{R \in \mathbb{R}^{2p\times k}}{\operatorname{argmin}} \mathbb{1}_{\mathbb{R}_+^{2p\times k}}(R) + \frac{\rho}{2}\|AX^{t+1} + R - E + Y^t\|_2^2 \tag{2.31b}$$

$$Y^{t+1} = Y^t + AX^{t+1} + R^{t+1} - E \ , \tag{2.31c}$$

where $V^t = \frac{\rho}{\eta} A^T(2Y^t - Y^{t-1})$. Then (2.31a) has a closed form solution based on element-wise soft-thresholding $X^{t+1} = \text{soft}(X^t - V^t, \frac{1}{\eta})$. The only problem left is how to compute $A^T Y$ in V^t, which can be solved as

$$A^T Y = \begin{bmatrix} \hat{C} & -\hat{C} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} - W_{\text{diag}} \ , \tag{2.32}$$

where $Y_1, Y_2 \in \mathbb{R}^{p\times k}$ are respectively upper and lower half of Y and $W_{\text{diag}} = W_{\text{diag1}} - W_{\text{diag2}}$. Assume the k-column block matrix X contains $(i+1)$-th column to $(i+k)$-th column in $\hat{B}$, then W_{diag1} is

$$W_{\text{diag1}} = \begin{bmatrix} \mathbf{0_1} \\ \mathbf{D}_{k\times k} \\ \mathbf{0_2} \end{bmatrix} \ , \tag{2.33}$$

where $\mathbf{0_1} \in \mathbb{R}^{i\times k}$ and $\mathbf{0_2} \in \mathbb{R}^{(p-i-k)\times k}$ are matrices with all zero entries. $\mathbf{D}_{k\times k} \in \mathbb{R}^{k\times k}$

is a diagonal matrix block, in which the m-th diagonal element $d_{m,m} = \tau_n Y_{1(m+1)}^T \mathbf{1}_p$ and $Y_{1(m)}$ is the m-th column of Y_1 in (2.32). W_{diag2} has the same format based on Y_2. The update of R^{t+1} can be done in closed form as $R^{t+1} = \max(H^t, 0)$, applied elementwise, where $H^t = E - Y^t - AX^{t+1}$.

For stage two, the problems for different columns only differ in the threshold λ_j in (2.24a), therefore the corresponding update in (2.25) can be done in element-wise parallel manner for column blocks.

Stability of Column-Block ACLIME-ADMM The inexact ADMM algorithm introduced two parameters, i.e., the scaled stepsize ρ and linearization parameter η. In (Wang and Banerjee, 2014), it is proved that the value of η depends on the convexity of the objective function. The experimental results for synthetic datasets show that the proposed algorithm is stable within the reasonable range of ρ. For a fixed η, the converge rate can be guaranteed if $\eta \geq \rho\lambda_{max}^2(C)$, where $\lambda_{max}(C)$ is the largest eigenvalue of covariance matrix. We validate the stability with a 10-variable synthetic dataset with 1500 samples, in which the variables follow multivariate Gaussian distribution. The underlying undirected graph is shown in Fig. 2.1(a). Fig. 2.1(b) and Fig. 2.1(c) show that the primal and dual residual converges to 0 and the estimated matrix can always detect the non-zero elements (i.e., the undirected edge in graph) correctly when ρ for both stages are chosen in $(0, 2]$. The proposed column-block ACLIME-ADMM can be achieved based on the parallel processing for separate column blocks, which leads to the high-efficiency and scalability. The estimation of precision matrix for large scale datasets is solvable with limited working memory.

2.4 PC Stable and Temporal Models

We use a variation of the classic *PC* algorithm as baseline algorithm for comparison. This section provides details of that algorithm and explains how structure learning algorithms can be used to derive temporal models.

2.4.1 PC Stable Algorithm

One of the best-known algorithms for structure learning is the well-known *PC* algorithm (Spirtes and Glymour, 1991). Colombo and Maathuis (Colombo and Maathuis, 2014) developed an improved version of the PC algorithm, called PC stable. PC stable is order-independent, more robust and easy to parallelize, and is used in this work. PC stable has only one parameter to choose, the significance value α for the statistical independence tests. We used $\alpha = 0.05$ for the runs with synthetic data and $\alpha = 0.1$ for the runs with observed data. There is generally little difference in the output of the PC stable algorithm for varying values of α (even up to $\alpha = 0.5$), so such a small change has no relevance for the results.

2.4.2 From Static to Temporal Model

Structure learning methods, including PC stable, CLIME-ADMM and ACLIME-ADMM, treat their input data as static data, i.e. the order of the samples does not matter. Most data in the geoscience, however, comes from temporal processes and the order of, and temporal distance between, samples is crucial for their interpretation. We can adapt structure learning algorithms to incorporate that information and to capture those temporal relationships explicitly using the approach first proposed by Chu et al. (Chu et al., 2005). The key idea is to introduce lagged variables into the model that capture the relationship between variables at different instances in time. The data of those lagged variables is populated from the original data and encapsulates the temporal information. In effect, we can thus turn a data set with q variables and temporal information into a data set with $p = (q \times T)$ variables, where T is the number of lagged copies for each variable. The new dataset can be treated as a static data set, and thus can be handled by standard structure learning algorithms. Once the static model with lagged variables is solved, the output can be converted to model the original variable set but complete with temporal relationships. The price to pay for this temporal model is high complexity, because rather than dealing with q variables, we deal with $p = (q \times T)$. This is another reason why we often encounter very high-dimensional problems in the geoscience. There are some associated initialization issues, but those can easily be overcome (Ebert-Uphoff and Deng, 2014). We adopt this approach for all algorithms used here. For the synthetic datasets (see Section 2.6), we have $q = 400, T = 20$, so that $p = 8,000$ with $n = 5,200$ samples; for the real dataset, $q = 800, T = 15$, so $p = 12,000$ with $n = 4,500$ samples. Note that since *ACLIME-ADMM* works with p^2 edges in each stage, the optimization for synthetic data involves 64 million variables and that for the real data involves 144 million variables.

2.5 Synthetic and Observed Datasets

2.5.1 Simulated Advection-Diffusion Processes

As a testbed for structure learning we created a simulation of a two-dimensional advection-diffusion process. This testbed generates synthetic data sets with known diffusion and advection properties, as benchmarks to test and compare different structure learning algorithms. We selected advection (e.g. transfer of heat through movement of a fluid) and diffusion (e.g. spread of heat in a resting fluid) processes, because in many geoscience applications they represent the two most dominant processes.

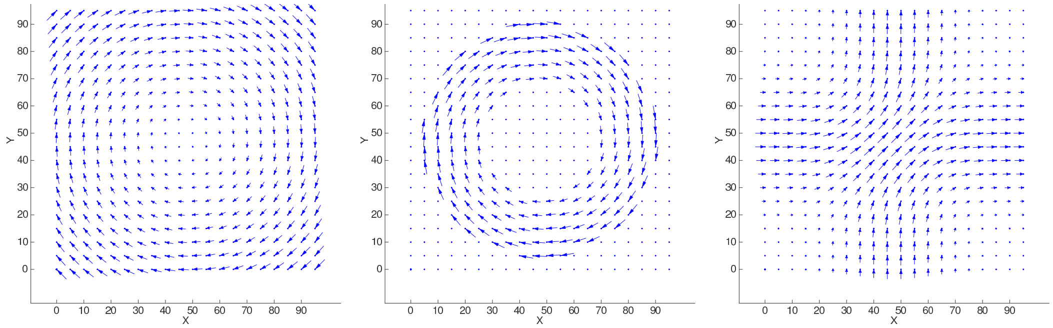
The two-dimensional advection-diffusion process is described by the following partial differential equation (PDE):

$$\frac{\partial f}{\partial t} + \left(V_x \frac{\partial f}{\partial x} + V_y \frac{\partial f}{\partial y} \right) = \left(\kappa_x \frac{\partial^2 f}{\partial x^2} + \kappa_y \frac{\partial^2 f}{\partial y^2} \right), \tag{2.34}$$

where $f(x, y, t)$ can be interpreted as the temperature of a fluid at location (x, y) over time t, κ_x and κ_y are the diffusion coefficients in x and y-direction, respectively, and $V(x, y)$ is the velocity vector field that describes the advection velocity at any point (x, y). For the results described here diffusion is symmetric, $\kappa_x = \kappa_y = \kappa$. We use a square grid with periodic boundary conditions, i.e. we apply a wrap-around in both x and y direction. To ensure that the connectivity between the grid points is encoded in the data, we send one signal to each grid point (one at a time) to disturb the system from equilibrium, let the signal travel to other points and dissipate, then repeat the process with the next grid point. Finally, we create three distinct scenarios for testing by choosing three different advection fields. In Scenario 1 (Fig. 2.2(a)) the advection field is circular and the magnitude of the velocity is proportional to the distance of the grid point from the grid center. Note that the velocity direction near the boundaries is discontinuous because of the wrap-around at the boundaries. Scenario 1 tests the effect of discontinuity. In Scenario 2 (Fig. 2.2(b)) the advection velocity is non-zero only in a ring shape. Inside and outside of that ring, advection velocities are zero, i.e., in those areas only diffusion is present. Scenario 2 can thus be used to test the algorithms for larger areas with only diffusion. In Scenario 3 (Fig. 2.2(c)) there are two crossing currents. One flows from left to right, the other from bottom to top. Advection velocities outside the main currents are small, but not zero.

2.5.2 Observed Data

We use data from the NCEP-NCAR reanalysis project (Kalnay et al., 1996a). The NCEP-NCAR reanalysis project provides data on a global grid for a variety of atmospheric variables and is derived from observations, but also incorporates the output of numerical weather predictions to improve the quality of the data. We use daily geopotential height data at 500mb, which denotes for any location the *height* at which the air pressure is 500mb. Data from the years 1950-2000 is used here, and, in order to focus on the dynamics of only one season only daily data from the boreal winter months (Dec, Jan, Feb) are used. Since irregularities in the grid, such as varying cell size, are known to create artifacts in the results of structure learning (Ebert-Uphoff and Deng, 2014), the data is interpolated on an 800-point grid of nearly equally distributed points, called *Fekete* points, on the sphere (Bendito et al., 2007).



(a) Scenario 1: Circular flow

(b) Scenario 2: Ring flow

(c) Scenario 3: Cross current

Figure 2.2: Advection Velocity Fields for the three scenarios, which can be interpreted as velocity of fluid flow at each point of square grid.

2.6 Experimental Results

In this section we compare results from the *PC stable*, *CLIME-ADMM* and *ACLIME-ADMM* methods for synthetic and real world data. We first discuss the experimental setup and implementation details.

2.6.1 Interpretation and Error Measures

The result of each structure learning algorithm is an adjacency matrix that describes which nodes in the graph are connected. Since we are learning a temporal model, each node in the graph represents a location (grid point) coupled with a specific time stamp. Thus each connection from the adjacency matrix represents a connection between two physical locations along with the two time stamps, so we can deduct the time it took to travel from potential source to potential effect. Connections with identical time stamps are interpreted as undirected edges. The remaining edges are directed, going from the location with the earlier time stamp to the one with the later time stamp. While it might be tempting to try to develop error measures directly for those edges (or for the corresponding adjacency matrices), those would be misleading. The reason is that physical connections do not have a *unique* representation in this space. For example, a signal that travels one grid point in one time step can be represented by a connection spanning one grid point distance in one time step, or by a connection spanning two grid point distances in two time steps, or both. More generally, there are many ways in which signal propagation can be represented in this framework, and methods should not be punished for using different, legitimate representations. The way to resolve this problem is *to focus on physically meaningful quantities*, since those are by definition unique. In this case a natural choice is to calculate an estimated velocity field, i.e. for each grid point we estimate a velocity vector by taking the average of all directed edges incident at the grid point, with each edge normalized by its travel time, T, which is the difference between the time stamps of its two end points. (We include both incoming and outgoing edges at each grid point to increase the robustness of the estimates.) This results in an estimated velocity vector at each grid point, which then can be compared directly to the advection velocities shown in Fig. 2.2.

Even if the structure learning method was perfect, we could not expect an exact match between the two fields—because of simulation errors and the fact that the advection field does not model the diffusion effects—but the results should be very similar to each other. Thus this is the best ground truth we can get for such a physical set-up.

Note that we can provide error measures only for the synthetic data, since the observed data does not have any quantitative ground truth. For the observed data we also generate velocity plots and compare them (visually) to domain knowledge in the geoscience.

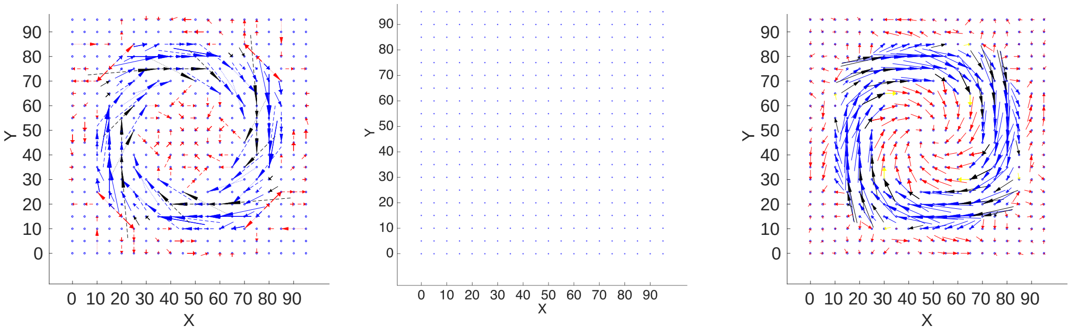
We use the following error measures. Numbering the grid points from $i = 1$ to 400, let L_i^{adv}, α_i^{adv} denote the length (magnitude) and angle of the advection velocity field at point i. $\hat{L}_i, \hat{\alpha}_i$ denote the corresponding velocity estimates obtained through structure learning. Then $\Delta\alpha_i = \text{abs}(\alpha_i^{\text{adv}} - \hat{\alpha}_i)$ denotes the absolute angle error and $\Delta L_i = \text{abs}(L_i^{\text{adv}} - \hat{L}_i)$ denotes the absolute length error at Point i. Note that if either the advection field or the approximation has zero velocity at a grid point, then length ΔL_i is still well defined, while angle $\Delta\alpha_i$ is undefined. Note that if the velocity is zero in *both* advection and estimated velocity, we set $\Delta\alpha_i = 0$.

We report the following error measures:

- RMSE-Length: The root mean square error of ΔL_i;

- RMSE-Angle: The root mean square error of $\Delta\alpha_i$, taking only points into account for which $\Delta\alpha_i$ is well defined.

- PPDL15: The percentage of points for which $\Delta\alpha_i \leq 15$ degrees, out of all points for which $\Delta\alpha_i$ is well-defined.



(a) PC stable (b) CLIME-ADMM: nothing found! (c) ACLIME-ADMM

Figure 2.3: Velocity estimates from three algorithms for Scenario 4.

Ideally, we want both RMSE measures to be small and the percentage value PPDL15 as close as possible to 100. From a geoscience viewpoint, the direction of connections is generally more important than the exact speed of signal travel, thus the angle-related measures are more important than the length-related measures. To highlight the angle accuracy in the velocity plots for synthetic data, arrows in these plots are colored based on their angle deviation, $\Delta\alpha_i$. The color code is as follows: blue for deviation of $[0, 15]$ degrees, black for $(15, 30]$ degrees, yellow for $(30, 45]$ degrees, and red for $(45, 180]$ degrees. Furthermore, if the input velocity is zero, and the output velocity is non-zero, then the deviation angle, and thus color, is undefined. In that case a small length of the output vector indicates a better match, so colors are chosen as follows in that case: blue for length of $[0, 0.1]$, black for length of $(0.1, 0.5]$, and red for a length of $(0.5, \infty)$.

2.6.2 Results for Synthetic Data

Fig. 2.3 shows the results for Scenario 4 for all three algorithms (compare to Fig. 2.2(b)). CLIME-ADMM fails miserably for the high-speed scenario (Fig. 2.3(b)), in fact it does not find a *single* connection, while both PC stable and ACLIME-ADMM provide good results (Fig. 2.3(a,c)). This failure was a primary reason for developing ACLIME-ADMM, namely to provide a scalable algorithm that can handle high-speed connections.

For the remaining scenarios CLIME-ADMM performed similarly to ACLIME-ADMM, thus we now focus on the comparison of ACLIME-ADMM and PC stable. Table 2.1 shows the error measures for PC stable and ACLIME-ADMM for four scenarios and Fig. 2.4 shows the results for Scenarios 1-3 from PC stable and ACLIME-ADMM. Both algorithms capture the basic shape of the corresponding velocity fields in Fig. 2.2. However, there are some

Table 2.1: Error measures of velocity estimates for synthetic data.

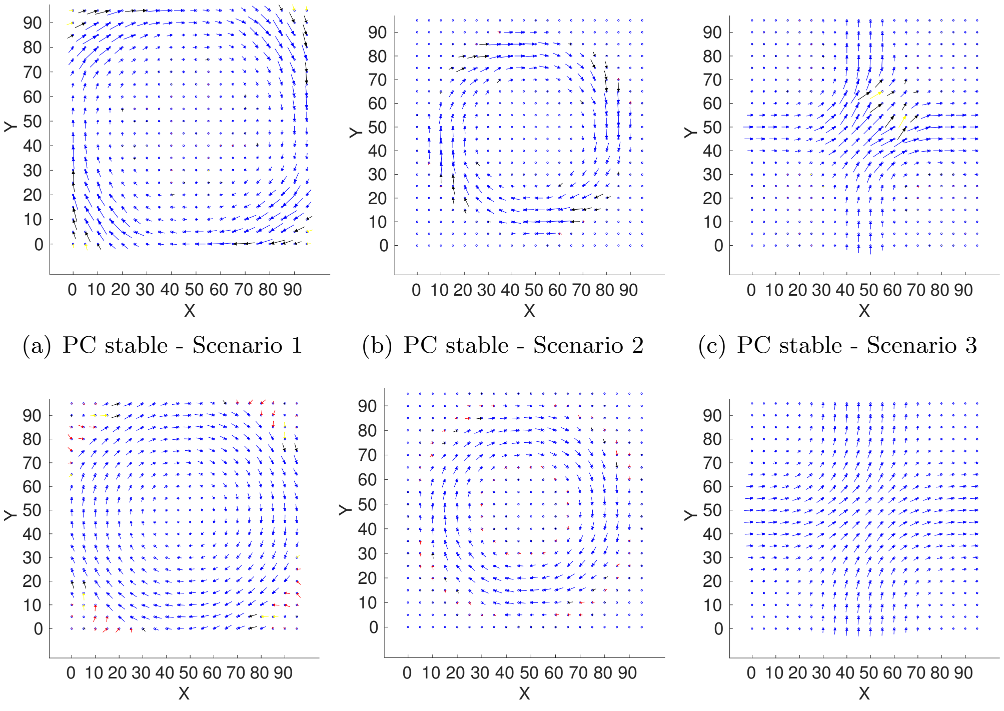
Scenario	Method	PPDL15	RMSE-Angle	RMSE-Length
Circular flow	PC stable	76	27.0671	0.8206
	ACLIME-ADMM	84	25.6995	0.7994
Ring flow	PC stable	90	11.2116	0.6241
	ACLIME-ADMM	83	7.1998	0.6124
Cross Current	PC stable	65.5	35.7746	0.7165
	ACLIME-ADMM	100	5.1364	0.7754
Fast Ring Flow	PC stable	49	59.8538	1.5234
	ACLIME-ADMM	30	50.0490	1.7871

differences. (1) ACLIME-ADMM tends to be more sensitive. Thus it is better than PC stable in identifying velocities of small magnitude (see center of Fig. 2.4(d)). (2) PC stable is better at handling contradicting edge directions near the boundary (see near the four corners in Fig. 2.4). (3) PC stable struggles with edges that do not align with the vertical or horizontal direction of the grid, i.e. diagonal directions tend to be distorted (strongest effect at center of Fig. 2.4(c), but the rings in Fig. 2.4(a, b) also appear more ”boxy”). ACLIME-ADMM does not show this problem, probably because of its higher sensitivity, i.e. the velocity estimates are calculated from a larger number of edges. Overall PC stable and ACLIME-ADMM both detect the primary patterns in the synthetic data, but ACLIME has generally higher accuracy than PC stable, as shown in Table 2.1, and is better at picking up weaker signals.

2.6.3 Results for Observed Data

Fig. 2.5(a-d) shows the velocities obtained from PC stable and ACLIME-ADMM for the dataset of observed daily geopotential height data for the Northern and Southern hemisphere. As a reference, we present the well known wind flow patterns in Fig. 2.6(a), as well as wind patterns at 500mb height in Fig. 2.6(b). The estimates are obtained in a similar way as for the synthetic data, but here only outgoing edges are considered at each node. Color indicates connectivity of the grid points: for each grid point we count the number of directed edges incident at that point, i.e. the number of edges contributing to its velocity estimate. This number indicates strength of connectivity at that point.

Firstly, ACLIME-ADMM shows even higher sensitivity for the observed data than for the synthetic data, resulting in a much larger number of arrows and higher connectivity than PC stable. Secondly, the results from both algorithms show information transfer mostly consistent with well known wind directions. Namely, the spatial distribution of winds at



(a) PC stable - Scenario 1 (b) PC stable - Scenario 2 (c) PC stable - Scenario 3

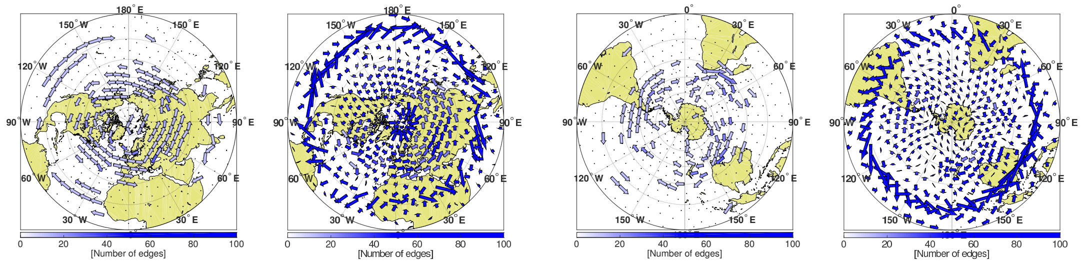
(d) ACLIME - Scenario 1 (e) ACLIME - Scenario 2 (f) ACLIME - Scenario 3

Figure 2.4: Velocity estimates from PC stable for Scenarios 1, 2 and 3 are decent, but some directions are distorted. Velocity estimates from ACLIME-ADMM for Scenarios 1, 2 and 3 are more accurate than PC stable.

500mb is such that easterlies (winds blowing from east to west) dominate the tropical bands (15S-15N), while westerlies (winds blowing from west to east) dominate mid latitudes (30N-60N), and another band of weak easterlies are typically seen in the polar region (Fig. 2.6(f)). Both algorithms capture the two major bands of easterlies and westerlies. However, the results from ACLIME-ADMM additionally detect very strong information flow near the equator, which cannot be readily explained by the weak easterlies seen at 500mb. We are currently exploring alternative explanations, such as these edges maybe being tied to weather features of similar lifecycles occurring simultaneously at different locations, etc.

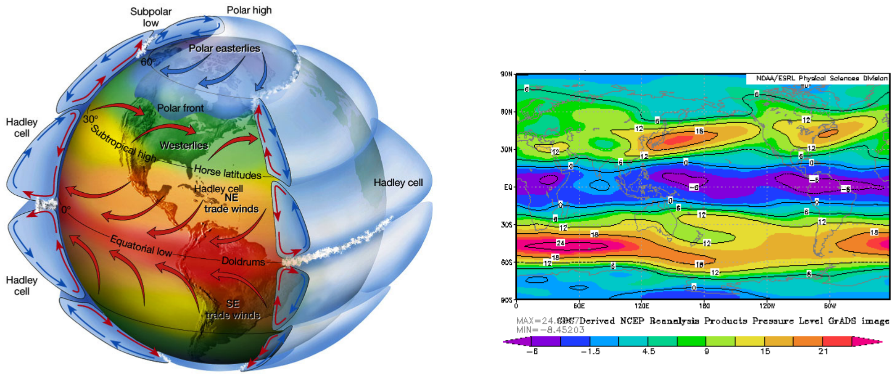
2.7 Conclusions

The main contribution of this chapter is a new algorithm, ACLIME-ADMM, which is suitable for high-dimensional structure learning and for small sample sizes. The work was motivated by geoscience applications, primarily the use of structure learning to identify interactions between different locations around the globe. PC stable was previously used for



(a) PC stable - North (b) ACLIME - North (c) PC stable - South (d) ACLIME - South

Figure 2.5: Velocity estimates in Northern ((a) and (b)) and Southern ((c) and (d)) Hemisphere.



(a) Circulation (b) Wind at 500mb

Figure 2.6: Atmospheric wind circulation: (a) global circulation patterns and surface winds; and (b) wind at 500 mb height (yearly average).

this application and is used here for comparison. PC stable gives decent, stable results, but is currently limited in the number of variables it can handle (about 100,000), which is not sufficient for many high-dimensional geoscience applications extending over both space and time. CLIME-ADMM, which promises to be much more scalable (already used for 1,000,000 variables for other applications), was applied for the first time to this application. It performed well for most scenarios, but failed miserably for the high speed signals (Scenario 4), where PC stable still gave good results. This motivated the development of the new algorithm, ACLIME-ADMM, which builds on CLIME-ADMM, but adjusts to local properties of the data. ACLIME-ADMM is much more sensitive than PC stable, thus produces denser plots, and is able to identify weaker signals. For the synthetic data ACLIME-ADMM provided the best overall results, including good results for the high-speed scenario. For observed data, both algorithms detect the strong easterlies and westerlies bands. Furthermore, ACLIME-ADMM yielded new strong edges near the equator that still need to be traced back to a specific physical mechanism. Clearly, more work needs to be done in order to fully understand the differences between the results obtained from CLIME-ADMM and PC stable. However, ACLIME-ADMM was shown to be a very promising candidate for structure learning in many climate science applications.

Chapter 3

Interpretable Predictive Modeling for Climate Variables with Weighted Lasso

3.1 Introduction

Over the past decade climate datasets with improved spatial resolutions have become available. While such datasets come from a mix of real observations and physics based models, recent years have seen considerable interest in applying machine learning techniques for predictive modeling of climate variables of interest. Such models have the potential to aid a better understanding of the impact of climate change and attribution of observed events as well as guide decision/policy making in a variety of domains such as agricultural planning, water resource management, and extreme weather events (O'Brien et al., 2006).

We consider one such problem in climate science of identifying predictive relationships between ocean sea surface temperature (SST) and land temperature (Steinhaeuser et al., 2011a). Recent work has shown sparse modeling techniques like Lasso (Chatterjee et al., 2012) tend to better capture predictive relationships between SST and land climate compared to more traditional methods like principal component regression (PCR) (Francis and Renwick, 1998), shallow neural networks (Steinhaeuser et al., 2011b), etc. From a climate science perspective, parsimony in variable selection leads to more interpretable models helping climate scientists gain a better understanding of the underlying relationships between climate variables. Still, there are difficulties in explaining the relationships due to the variable selection inconsistency of Lasso and the high spatial correlation among SST variables.

Inspired by the adaptive Lasso (Zou, 2006), we propose a weighted ℓ_1 regularized model suitable for spatial problems since it encourages the estimator to pick spatially contiguous SST covariates. The weighted ℓ_1 regularizer penalizes different components of regression coefficients θ differently and is mathematically defined by $R(\theta) = \sum_{i=1}^{p} w_i|\theta_i|$, where w_i is the weight for component i. Lower the weight, lower is the penalization on the corresponding covariates and consequently more are the chances they will be nonzero. Note that, adaptive Lasso is weighted Lasso, where the weights are chosen to be inversely proportional to the estimated coefficients from estimator like ordinary least squares (OLS). For the problem we consider, we propose the weights on ocean locations are directly proportional to their distance from the land location thus penalizing faraway ocean regions more, which is consistent with domain knowledge in climate science. We show that the weighted Lasso, in contrast to Lasso, gives more interpretable results which conform to the observations of nearby ocean locations having the most effect on land temperature.

We perform extensive comparison of the weighted Lasso with baselines on data from 3 different Earth System Models (ESMs) (Taylor et al., 2012). First, comparisons between weighted Lasso and Lasso shows that they achieve similar predictive performance, but weighted Lasso is considerably more interpretable in terms of variable selection. Second, somewhat surprisingly, we illustrate that weighted Lasso persistently outperforms Deep nets which form the state-of-the-art in many other application areas (He et al., 2016; Krizhevsky et al., 2012b; LeCun et al., 2015); weighted Lasso is also illustrated to have superior performance over gradient boosted trees (Chen and Guestrin, 2016) and PCR (Jolliffe, 2011). We also present a detailed analysis of the poor performance of Deep nets and report results on a variety of settings such as number of layers, number of hidden units, mini-batch size, regularization type, etc. The key factor limiting the performance is sample size. Deep nets overfit the training set leading to poor validation/test performance. The results emphasize the need for caution and further work on Deep nets for small sample (scientific) problems.

Our main contributions are as follows:

1. We suggest the weighted Lasso estimator, which incorporates domain knowledge for finding relationships in spatial data. We also derive non-asymptotic parameter estimation error bounds for the weighted Lasso estimator.

2. We show that weighted Lasso achieves high prediction accuracy and consistent variable selection for land climate prediction using SST compared to other latest state-of-the-art machine learning methods like Deep nets and gradient boosted trees.

3. We perform extensive experiments with Deep nets and show that Deep nets easily

overfit the training data without sufficient samples.

Organization of the chapter: We start with a discussion on related work. We then give finite sample estimation error bounds for weighted Lasso. We subsequently present experimental results comparing the weighted Lasso with baseline methods along with in-depth results on Deep nets.

3.2 Related Work

We briefly review the statistical models used in the climate science to discover predictive relationships between climate variables. Most statistical models perform some form of dimensionality reduction due to the large spatial datasets and relatively fewer data samples. A popular method is principal component regression (PCR) (Olivieri, 2018), which has been used for temperature and precipitation prediction in New Zealand (Francis and Renwick, 1998). In (Hsieh and Tang, 1998), principal component analysis (PCA) is used to compress large spatial fields followed by fitting a neural network on the compressed dataset. In (Steinhaeuser et al., 2011a) clustering is used for dimension reduction followed by various regression methods, such as linear regression, support vector regression, regression trees, to predict land temperature and precipitation from global SST field. In contrast (Chatterjee et al., 2012) model the same problem in (Steinhaeuser et al., 2011a) as a high-dimensional sparse regression problem where the land climate is the dependent variable, SST field are the independent variables and a sparsity promoting regularizer captures the constraint that land temperature is influenced by only a few ocean locations. More recently a spectral nonlinear dimensionality reduction method is used in (DelSole and Banerjee, 2017) to capture the relationship between summer Texas area temperature and Pacific SST.

Geostatistical methods, like kriging (Goovaerts, 1999) and its variations have been applied for spatial interpolation of climate variables (Aalto et al., 2013). However, such methods usually only perform well within a defined local neighborhood (Walter et al., 2001). Morevover the success of such methods relies on proper choice of kernels and hyperparameters which is statistically and computationally challenging in high-dimensional datasets.

There is increasing interest in exploring the application of Deep nets in climate applications inspired by their success in domains like image processing (He et al., 2016), speech recognition (LeCun et al., 2015), etc. Recent work explore the use of Deep nets for prediction of the Oceanic Niño Index (ONI) (McDermott and Wikle, 2017) and for statistical downscaling of climate variables (Vandal et al., 2017), although there is currently lacking an understanding or comprehensive study on the generalization performance of Deep nets on small sample size datasets routinely found in climate science applications.

3.3 Estimation Error Bound for Weighted Lasso

For land climate prediction using SST, the spatial information can be considered while designing the predictive models. Since land temperature is known to be mostly influenced by nearby ocean locations, we propose a modification of the weighted ℓ_1 regularizer used in weighted Lasso. It penalizes differently for temperature at each ocean location based on their distance from land target region.

In this section, we provide the non-asymptotic estimation error bound for the following weighted Lasso estimator in a general setting,

$$\hat{\theta} = \text{argmin}_{\theta \in \mathbb{R}^p} \frac{1}{2n} \|y - X\theta\|_2^2 + \lambda \sum_{i=1}^{p} w_i \, |\theta_i| \tag{3.1}$$

where, for our application, $y \in \mathbb{R}^n$ is the land temperature, $X \in \mathbb{R}^{n \times p}$ are SST at p ocean locations, $\theta \in \mathbb{R}^p$ are regression coefficients, θ_i is the ith coefficient in θ, w_i is the positive weight corresponding to θ_i, and λ is penalty parameter. The weights can be assigned in a data-dependent way or chosen intelligently using prior knowledge. For example, in our application, the weights $w_i, \ 1 \leq i \leq p$ are assigned to be proportional to the distance of the ocean location from the land location.

The weighted Lasso estimator is equivalent to the adaptive Lasso estimator (Zou, 2006), except for the procedure used to define the weights $w_i, \ 1 \leq i \leq p$. While prior work has focused on analysis of the adaptive Lasso estimator in the asymptotic setting (Huang et al., 2008; Zou, 2006), we derive results for the weighted Lasso estimator in the non-asymptotic setting. The results can also be suitably extended to the adaptive Lasso estimator.

Assumptions: Consider data generated according to the linear model $y_i = \langle x_i, \theta^* \rangle + \epsilon_i, \ 1 \leq i \leq n$ and θ^* is estimated using the weighted Lasso estimator. The rows of the design matrix $X \in \mathbb{R}^{n \times p}$ are independent sub-Gaussian random vectors with sub-Gaussian norm bounded by L and covariance matrix $\Sigma = E[x_i x_i^T]$. The noise $\epsilon_i \in \mathbb{R}, \ 1 \leq i \leq n$ is mean-zero i.i.d. sub-gaussian noise with sub-Gaussian norm less than 1. Assume the following,

1. The true parameter θ^* is s-sparse. Let $w^{\uparrow}$ denote the weight vector with elements in ascending order. We assume that the weights corresponding to the s non-zero elements in θ^* are among the smallest m weights $w^{\uparrow}_{1:m}$ in w. Also, let $\hat{\theta} = \theta^* + \Delta = \theta^* + \mathcal{M}(\Delta) + \mathcal{M}^{\perp}(\Delta)$, where $\Delta = \hat{\theta} - \theta^*$ is the error vector, $\mathcal{M}$ is the subspace, to which the m elements in θ^* corresponding to the weights $w^{\uparrow}_{1:m}$ belongs, and $\mathcal{M}^{\perp}$ is the orthogonal subspace.

2. When penalty parameter λ satisfies

$$\lambda \geq c' * \max\left\{\frac{\sqrt{m}}{\sqrt{n}\|w^{\uparrow}_{1:m}\|_2}, \frac{\sqrt{\log p}}{\sqrt{n}\tilde{w}_{\min}}\right\} , \tag{3.2}$$

where $c' > 0$ is a constant,and $\tilde{w}_{\min}$ is the minimum element in $\mathcal{M}^{\perp}(w^{\uparrow})$, the error set is

$$E_r = \{\Delta \in \mathbb{R}^p | R(\mathcal{M}^{\perp}(\Delta)) \leq \beta \|w^{\uparrow}_{1:m}\|_2 \|\mathcal{M}(\Delta)\|_2\} , \tag{3.3}$$

where $\beta > 1$ is a constant and $R(\theta) = \sum_{i=1}^{p} w_i |\theta_i|$. The restricted eigenvalue (RE) condition (Bickel et al., 2009) is assumed to be satisfied.

Theorem 2. *Under the above assumptions, the following bound holds on the error vector $\Delta = \hat{\theta} - \theta^*$ with high probability for some positive constant c,*

$$\|\Delta\|_2 \leq \frac{c}{\sqrt{n}}\left(\sqrt{m} + \frac{\|w^{\uparrow}_{1:m}\|_2\sqrt{\log p}}{\tilde{w}_{\min}}\right) . \tag{3.4}$$

Remark 1. *For Lasso $w_i = 1,\ 1 \leq i \leq p$. Hence in the context of the above result we recover the non-asymptotic estimation error of Lasso (Bickel et al., 2009; Chandrasekaran et al., 2012; Negahban et al., 2012) by substituting $m = s$, $\|w^{\uparrow}_{1:m}\|_2 = \sqrt{s}$ and $\tilde{w}_{\min} = 1$.*

Remark 2. *If the s lowest weights in $w^{\uparrow}$ correspond to the non-zero weights in θ^* then we note that $m = s$ and $\|w^{\uparrow}_{1:s}\|_2 / \tilde{w}_{\min} \leq \sqrt{s}$ thus giving an improvement over the corresponding bound for Lasso.*

Remark 3. *If we end up assigning the largest weights to the non-zero elements in θ^* then $m = p$ and we recover the bound $\|\Delta\|_2 \leq \sqrt{p/n}$ which is equivalent to performing ordinary least squares on the dataset.*

The weighted Lasso problem can be numerically optimized by converting it to a Lasso problem by rescaling the data with the weights (Zou, 2006).

3.4 Land Temperature Prediction

We analyze relationships between land temperature and SST in Earth system model (ESM) data. ESMs are numerical models representing physical processes in the ocean, cryosphere and land surface with data generated using simulations with different initial conditions (Pachauri et al., 2014; Taylor et al., 2012).

We use data from the historical runs of 3 ESMs (see Table 3.1) included as part of the core set of experiments in CMIP5 (Taylor et al., 2012). The historical runs of CMIP5 ESMs try to replicate observed climate conditions from 1850-2005 closely, capturing effects from changes in atmospheric CO_2 due to both anthropogenic and volcanic influences, solar forcing, land use, etc. Each monthly ESM dataset has SST data over a $2.5° \times 2.5°$ resolution grid of earth and corresponding monthly surface temperature over land locations. In effect for each ESM we have 1872 data points with 5881 ocean locations. Brazil, Peru, and Southeast Asia are selected as the 3 land target regions to study in this work as they are known to have diverse geological properties (Steinhaeuser et al., 2011b).

3.4.1 Experiment Setting

We divide the data into 10 training sets by applying a moving window of 100 years with a stride of 5 years. The 10 years subsequent to the end of the training set are used for testing. We deseasonalize each training-test set combination separately by z-scoring each month data with the corresponding monthly mean and standard deviation. Note that both train and test sets are z-scored using monthly means and standard deviations computed from the training set. We compare the performance of weighted Lasso against the following baseline methods:

1. ℓ_1 penalized least squares (**Lasso**) (Tibshirani, 1996): This is equivalent to setting all weights in weighted Lasso equal to 1.
2. Principal Component Regression (**PCR**) (Jolliffe, 2011): A popular method in climate science where principal components computed from training data are considered as covariates for ordinary least square regression on response variables.

Table 3.1: Description of the Earth System Models used in the experiments.

Model name	Origin	References
CMCC-CESM	Centro Euro-Mediterraneo per I Cambiamenti Climatici (Italy)	(Fogli et al., 2009), (Vichi et al., 2011)
INM-CM4	Institute for Numerical Mathematics (Russia)	(Volodin et al., 2010)
MIROC5-r1i1p1	Atmosphere and Ocean Research Institute, National Institute for Environmental Studies, Japan Agency for Marine-Earth Science and Technology, University of Tokyo (Japan)	(Watanabe et al., 2010)

3. Gradient Boosted Trees (**GBT**) (Chen and Guestrin, 2016): An ensemble method which uses decision tree as its weak learner. GBTs are implemented in Python using **xgboost** package (Chen and Guestrin, 2016).

4. Deep neural networks (**Deep nets**) (LeCun et al., 2015): Multilayer perceptrons with many hidden layers and CNNs. All networks are implemented in Python using **Keras** package (Chollet, 2015).

Table 3.2: Comparison of RMSE on test sets for land climate prediction of Brazil, Peru and South-east Asia using weighted Lasso and other baseline methods. Average RMSE ± standard error on test sets are shown. The minimum average RMSE for each target region is shown as bold. Weighted Lasso achieves overall best performance. Furthermore, linear model weighted Lasso and Lasso both outperform Deep nets and GBT.

Model	Location	Weighted Lasso	Lasso	PCR	Deep nets	GBT
CMCC -CESM	Brazil	$\mathbf{0.6580 \pm 0.0344}$	0.6681 ± 0.0317	0.8629 ± 0.0654	0.8151 ± 0.0364	0.7354 ± 0.0377
	Peru	$\mathbf{0.6901 \pm 0.0269}$	0.7120 ± 0.0214	0.8541 ± 0.0560	0.8476 ± 0.0283	0.7400 ± 0.0251
	SE Asia	$\mathbf{0.5217 \pm 0.0103}$	0.5284 ± 0.0109	0.7252 ± 0.0544	0.7424 ± 0.0153	0.5774 ± 0.0171
INM -CM4	Brazil	$\mathbf{0.7641 \pm 0.0173}$	0.7753 ± 0.0161	1.2030 ± 0.0637	0.9144 ± 0.0304	0.8707 ± 0.0193
	Peru	$\mathbf{0.7127 \pm 0.0144}$	0.7202 ± 0.0101	1.1879 ± 0.0654	0.8785 ± 0.0228	0.8164 ± 0.0184
	SE Asia	$\mathbf{0.7719 \pm 0.0171}$	0.7742 ± 0.0174	1.2751 ± 0.0645	0.9986 ± 0.0290	0.8970 ± 0.0218
MIROC5 -r1i1p1	Brazil	$\mathbf{0.5395 \pm 0.0258}$	0.5614 ± 0.0266	0.7214 ± 0.0566	0.6822 ± 0.0263	0.6005 ± 0.0279
	Peru	$\mathbf{0.5441 \pm 0.0331}$	0.5764 ± 0.0350	0.7654 ± 0.0581	0.6979 ± 0.0227	0.5953 ± 0.0262
	SE Asia	$\mathbf{0.5092 \pm 0.0154}$	0.5308 ± 0.0164	0.8139 ± 0.0712	0.7500 ± 0.0187	0.5758 ± 0.0098

Table 3.3: Comparison of R^2 on test sets for land climate prediction of Brazil, Peru and South-east Asia using weighted Lasso and other baseline methods. Average R^2 ± standard error on test sets is shown. The maximum average R^2 for each target region is shown as bold. Weighted Lasso achieves overall best predictive performance. Furthermore, linear model weighted Lasso and Lasso both outperform Deep nets and GBT.

Model	Location	Weighted Lasso	Lasso	PCR	Deep nets	GBT
CMCC -CESM	Brazil	$\mathbf{0.4887 \pm 0.0555}$	0.4697 ± 0.0595	0.1372 ± 0.0982	0.2292 ± 0.0633	0.3706 ± 0.0587
	Peru	$\mathbf{0.4044 \pm 0.0357}$	0.3655 ± 0.0333	0.1004 ± 0.0704	0.1018 ± 0.0476	0.3168 ± 0.0336
	SE Asia	$\mathbf{0.6963 \pm 0.0205}$	0.6901 ± 0.0188	0.4086 ± 0.0763	0.3763 ± 0.0509	0.6312 ± 0.0222
INM -CM4	Brazil	$\mathbf{0.1855 \pm 0.0342}$	0.1616 ± 0.0340	-1.098 ± 0.2387	-0.1650 ± 0.0639	-0.0629 ± 0.0550
	Peru	$\mathbf{0.3457 \pm 0.0324}$	0.3334 ± 0.0258	-0.8853 ± 0.2447	-0.0034 ± 0.0680	0.1372 ± 0.0498
	SE Asia	$\mathbf{0.3131 \pm 0.0262}$	0.3091 ± 0.0266	-0.8827 ± 0.1583	-0.1498 ± 0.0568	0.0727 ± 0.0372
MIROC5 -r1i1p1	Brazil	$\mathbf{0.7615 \pm 0.0369}$	0.7413 ± 0.0390	0.5878 ± 0.0616	0.6263 ± 0.0423	0.7146 ± 0.0330
	Peru	$\mathbf{0.7609 \pm 0.0300}$	0.7331 ± 0.0326	0.5120 ± 0.0843	0.5964 ± 0.0503	0.7186 ± 0.0256
	SE Asia	$\mathbf{0.7436 \pm 0.0298}$	0.7224 ± 0.0309	0.2949 ± 0.1448	0.4584 ± 0.0409	0.6794 ± 0.0260

The models are evaluated quantitatively on test sets based on two metrics: (a) the root mean square error (RMSE), defined as RMSE $= \sqrt{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2/n}$; (b) the coefficient of the determination (R^2), given by $R^2 = 1 - \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 / \sum_{i=1}^{n}(y_i - \bar{y})^2$, where for the i-th data point, y_i is the true normalized land temperature for a target region and $\hat{y}_i$ is the corresponding estimated value. $\bar{y}$ is the average value for all n data points. The hyperparameters for weighted Lasso (regularization parameter), Lasso (regularization parameter), PCR (number of principal components for regression) and GBT (learning rate and maximum depth of tree) are selected by validation set. Specifically, in each training set we select the first 80 years to train the model and use the next 20 years as a validation set. The hyperparameters giving best performance on the validation set are chosen. We then refit the predictive models on the full training set using the chosen hyperparameters. For GBT, we fix the number of trees to 100, and perform a grid-search to find the optimal learning rate and maximum depth of tree. For all models the optimal value of learning rate on the validation set varies between 0.05 and 0.07 and the optimal maximum tree depth is found to be 3. For Deep nets we experiment with various combinations of: (a) the number of hidden layers, (b) the number of hidden units in each layer, (c) different mini-batch size when training using the Adam optimization algorithm (Kingma and Ba, 2014), and (d) ℓ_1, ℓ_2 and no regularization. Each network uses Relu (Nair and Hinton, 2010) as activation function. The maximum number of epochs for training is set as 150. We also use early-stopping by examining validation set error. In almost all cases, an 8 hidden layer Deep nets with ℓ_1 regularization on the weights gave the best performance on the validation set. We report results with mini batch size set to 32. We also run experiments with transfer learning (Yosinski et al., 2014) for Convolutional Neural Networks (CNN) (Lecun et al., 1998) by training only the last two layers of the Resnet-50 (He et al., 2016) which is pre-trained on ImageNet (Russakovsky et al., 2015). Resnet-50 is found to have worse performance in comparison to Deep nets and hence, in the interest of brevity and space, we exclude it from the comparison. More details on the performance of Resnet-50 can be found in Table 3.4.

Table 3.4: Comparison of the best RMSEs among weighted Lasso, Deep nets, and Resnet-50 using data from CMCC-CESM. Resnet-50 shows the worst predictive accuracy.

Location	Weighted Lasso	Deep nets	Resnet-50
Brazil	$\mathbf{0.6513 \pm 0.0635}$	0.8151 ± 0.0364	1.2972 ± 0.4109
Peru	$\mathbf{0.6944 \pm 0.0444}$	0.8476 ± 0.0283	1.3739 ± 0.3211
SE Asia	$\mathbf{0.5162 \pm 0.0213}$	0.7424 ± 0.0153	1.4760 ± 0.4227

3.4.2 Experimental Results

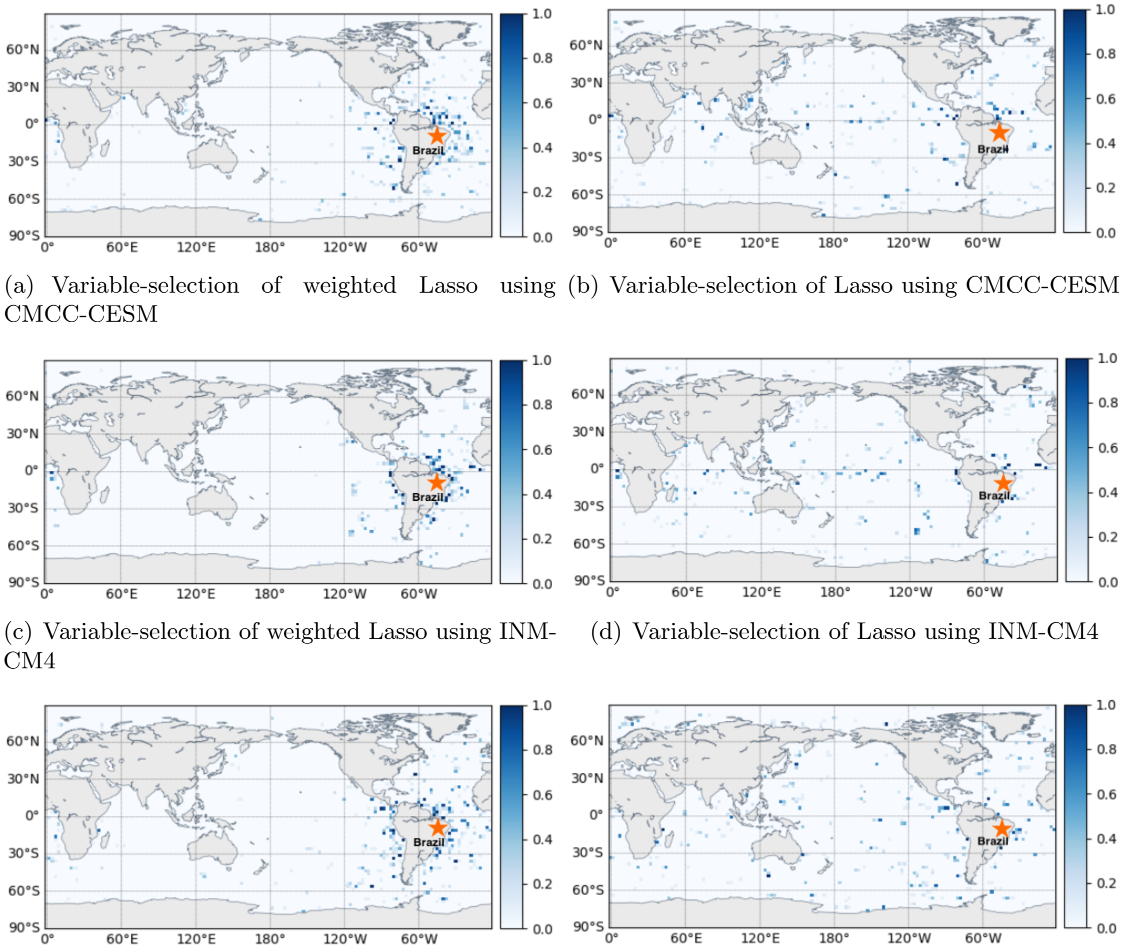
We compare different baseline methods against weighted Lasso using average RMSE and R^2 over test sets. We also show an in-depth analysis of the performance of Deep nets for our application.

Prediction Accuracy Table 3.2 and Table 3.3 report the average RMSE, R^2, and their standard errors. Weighted Lasso achieves better average predictive accuracy compared to other baseline methods across all 3 ESMs. The p-values of 2-sample K-S test (Daniel, 1978) for RMSE on test sets are shown in Table 3.5. Weighted Lasso is significantly better than PCR, Deep nets and GBT ($p < 0.05$) in most cases (21 out of 27). While the prediction accuracy of weighted Lasso is not significantly better than Lasso, we show that weighted Lasso consistently chooses a subset of variables of which ocean locations are close to the land target region, which is more interpretable in climate science perspective.

Table 3.5: The p-values from 2-sample KS-test on RMSE of test sets of weighted Lasso against other baseline methods are shown. The p-values less than 0.05 are shown in bold. The performance of weighted Lasso is significantly better than non-linear baseline methods for most of target regions.

Model	Location	Lasso	PCR	Deep nets	GBT
CMCC-CESM	Brazil	0.9747	0.1108	**0.0310**	0.1108
	Peru	0.6750	0.3128	**0.0068**	0.3128
	SE Asia	0.6750	**0.0001**	**0.0000**	**0.0068**
INM-CM4	Brazil	0.6750	**0.0001**	**0.0012**	**0.0012**
	Peru	0.9747	**0.0000**	**0.0000**	**0.0068**
	SE Asia	0.9747	**0.0000**	**0.0001**	**0.0068**
MIROC5-r1i1p1	Brazil	0.9747	**0.0339**	**0.0120**	0.1473
	Peru	0.6750	0.1108	**0.0120**	0.3743
	SE Asia	0.6750	**0.0000**	**0.0000**	**0.0120**

Variable selection Weighted Lasso and Lasso introduce sparsity in variable selection. During the training phase, an ocean location is considered selected, if it has a corresponding non-zero coefficients. The behavior of weighted Lasso (and Lasso respectively) is similar for land climate prediction across 3 land target regions. We analyze the ocean locations selected by Lasso and weighted Lasso for Brazil temperature prediction for all the 10 runs for all ESM models as an example. Figure 3.1 plots for each ESM model the number of times each location is selected across the 10 runs. We make two observations from the plots: (a) weighted Lasso assigns non-zero weights to ocean location close to Brazil consistent with domain knowledge, and (b) variable selection in weighted Lasso is more stable compared to Lasso in the sense that the same locations are picked in all 10 datasets. However, Lasso has few variables which are consistently chosen in all predictive models for the same land target region. Also, the frequently selected variables using Lasso are distributed at arbitrary locations, which is not interpretable in climate science perspective.



(a) Variable-selection of weighted Lasso using CMCC-CESM

(b) Variable-selection of Lasso using CMCC-CESM

(c) Variable-selection of weighted Lasso using INM-CM4

(d) Variable-selection of Lasso using INM-CM4

(e) Variable-selection of weighted Lasso using: MIROC5

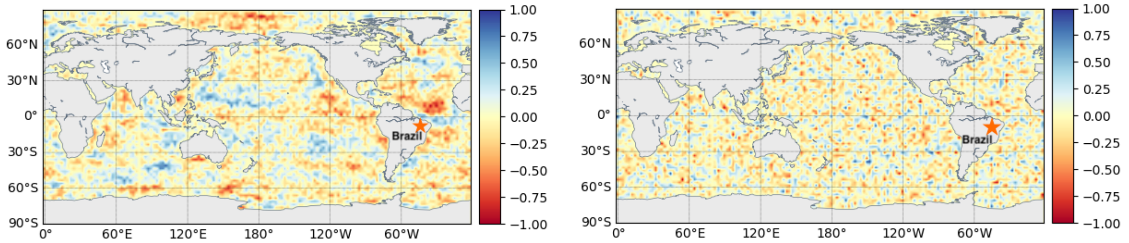
(f) Variable-selection of Lasso using: MIROC5

Figure 3.1: Comparison of variable selection by Lasso and weighted Lasso for Brazil temperature prediction. The plot shows the probability that each ocean location is selected in the 10 runs for each ESM model. In contrast to Lasso, weighted Lasso chooses more ocean locations closer to Brazil and achieves more consistent variable selection.

We also compare the weights from a unit from the first layer in Deep nets in Figure 3.2. Deep nets assign non-zero weights for all ocean locations even with ℓ_1 regularization.

3.4.3 Deep Nets: What Happened?

In this section, we analyze various facets of the performance of Deep nets. The performance of Deep nets is influenced by the number of hidden layers, number of hidden units, mini-batch size, regularization etc. We analyze the impact of each of these on the performance of Deep nets by varying one of the parameters while keeping the others fixed. We also demonstrate that Deep nets overfit the training data and hence do not generalize well on the test set.



(a) Weights of Deep nets without regularization.

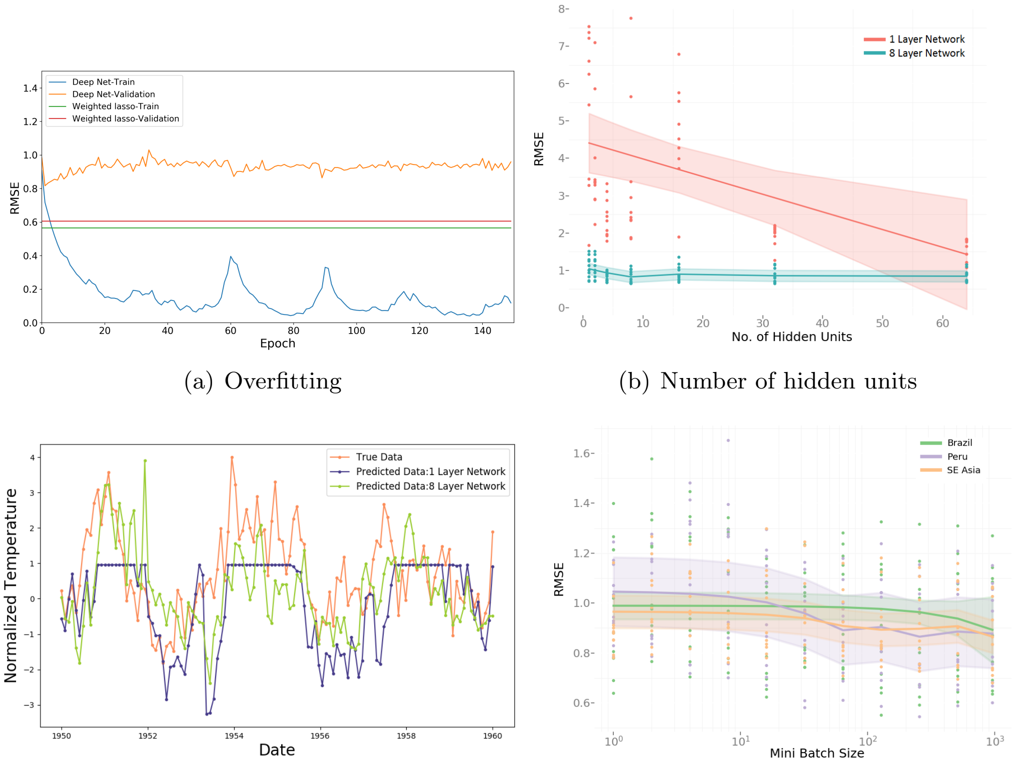
(b) Weights of Deep nets with ℓ_1 regularization.

Figure 3.2: Comparison of regression coefficients of a unit from Deep nets with and without ℓ_1 regularization for Brazil. All weights are normalized to $[-1, 1]$ by dividing the largest value among absolute weights.

Overfitting Figure 3.3(a) shows the training and validation set RMSE after each epoch for a 8 layer Deep nets with 32 hidden units trained for temperature prediction over Brazil. The Deep nets training error stabilizes after about 20 epochs and is lower than the RMSE of linear models. In contrast the validation set error of the Deep nets is much higher which indicates that Deep nets overfit the noise in the training set and hence can not generalize well over the unseen test set.

Effect of number of hidden units Figure 3.3(b) plots the test RMSEs for temperature prediction over Brazil as we alter the number of hidden units in each layer. The RMSE slightly decreases as the number of hidden units increases from 1 to 64 for both shallow networks with 1 hidden layer, and Deep nets with 8 hidden layers.

Shallow vs Deep Structure Figure 3.3(c) compares Deep nets with 1 hidden layer against Deep nets with 8 hidden layers on test set prediction over Brazil. Having more layers gives better test set RMSE.



(a) Overfitting

(b) Number of hidden units

(c) Shallow vs. deep

(d) Mini batch size

Figure 3.3: (a) An example of model overfitting for Deep nets. Deep nets are trained for 150 epochs. The blue curve and orange curve indicate the RMSE of the training and validation set for Deep nets. There is a clear gap between the training and validation RMSE. The RMSE of weighted Lasso on both training (green line) and validation (red line) sets are also shown for comparison. (b) Average RMSE on test sets vs. number of hidden units for Brazil temperature prediction with CMCC-CESM. The shaded zone indicates the confidence intervals (95%) around the predicted mean. For both 1-Layer and 8-Layer network configuration, the RMSE tends to slightly decrease with increasing number of hidden units. (c) The comparison of predicted land temperatures in Brazil with CMCC-CESM over a 10 year period (1950-1960) between a shallow and a deep network structure. The deep structure predictions are better than a shallow network. (d) Average test RMSE vs mini batch size over Brazil, Peru, and SE Asia for CMCC-CESM. Mini batch size of 1, 2, 4, 8, 16, 32, 64, 128, 256, 512 as well as full batch size are used on a 8 hidden layer network. The average RMSE on test sets decreases as batch size increases.

Effect of mini-batch size Mini-batch size while training is believed to have a strong impact on Deep nets performance (Bengio, 2012; Masters and Luschi, 2018). We analyze the effect on average test RMSE of mini-batch size for temperature prediction over all three land locations (Figure 3.3(d)). The RMSE are highest with small batch sizes, steadily decreasing with increasing batch size.

Effect of Regularization We explore 3 regularization schemes, ℓ_1, ℓ_2 and $\ell_1+\ell_2$. Table 3.6 shows the comparison on the test RMSE values of weighted Lasso and Deep nets before and after applying ℓ_1, and ℓ_2 regularization. ℓ_1 regularization seems to give better performance over other regularization schemes including no regularization.

Table 3.6: Comparison of RMSE on test sets of regularized Deep nets and weighted Lasso. Average test RMSE ± standard error are shown. Deep nets with ℓ_1 regularization has smaller test set RMSE than ℓ_2 and $\ell_1 + \ell_2$ regularization.

Model	Location	Weighted Lasso	Deep nets	Deep nets with ℓ_1	Deep nets with ℓ_2	Deep nets with $\ell_1 + \ell_2$
CMCC -CESM	Brazil	$\mathbf{0.6580 \pm 0.0344}$	0.8151 ± 0.0364	0.6931 ± 0.0260	0.8458 ± 0.0744	1.0635 ± 0.1308
	Peru	$\mathbf{0.6901 \pm 0.0269}$	0.8476 ± 0.0283	0.7499 ± 0.0210	0.9099 ± 0.0374	1.2099 ± 0.0453
	SE Asia	$\mathbf{0.5217 \pm 0.0103}$	0.7424 ± 0.0153	0.5656 ± 0.0063	0.6869 ± 0.0215	0.8992 ± 0.0324
INM -CM4	Brazil	$\mathbf{0.7641 \pm 0.0173}$	0.9144 ± 0.0304	0.8453 ± 0.0185	0.7334 ± 0.0403	1.0992 ± 0.1232
	Peru	$\mathbf{0.7127 \pm 0.0144}$	0.8785 ± 0.0228	0.7815 ± 0.0119	0.7193 ± 0.0390	1.1686 ± 0.0964
	SE Asia	$\mathbf{0.7719 \pm 0.0171}$	0.9986 ± 0.0290	0.8585 ± 0.0204	0.7891 ± 0.0600	1.2414 ± 0.1042
MIROC5 -r1i1p1	Brazil	$\mathbf{0.5395 \pm 0.0258}$	0.6822 ± 0.0263	0.5919 ± 0.0160	0.9884 ± 0.0278	1.2544 ± 0.0477
	Peru	$\mathbf{0.5441 \pm 0.0331}$	0.6979 ± 0.0227	0.5848 ± 0.0358	0.8781 ± 0.0177	1.5395 ± 0.1526
	SE Asia	$\mathbf{0.5092 \pm 0.0154}$	0.7500 ± 0.0187	0.5616 ± 0.0194	0.9887 ± 0.0458	1.3306 ± 0.0949

3.5 Conclusions

In this chapter, we propose a weighted Lasso scheme for prediction on spatial climate data in order to encode the inherent spatial information in such datasets. Also, the non-asymptotic estimation error bound for weighted Lasso is given. The proposed method is evaluated on a task to predict temperature for 3 distinct land target regions using SST from the historical runs of 3 ESMs. The weights are set to be proportional to the geographical distance between the ocean location of each predictor and the target land region, constraining the estimator to pick spatially nearby ocean locations. Weighted Lasso not only achieves better prediction accuracy compared to other linear and non-linear models, including PCR, GBT and Deep nets across all ESMs, but also selects stable predictors consistent with domain knowledge.

We also conduct a comprehensive analysis of Deep nets on high-dimensional climate datasets with small sample size. Empirical results show that linear models outperform the non-linear models and thus are more suitable for climate problems where the number of samples is limited.

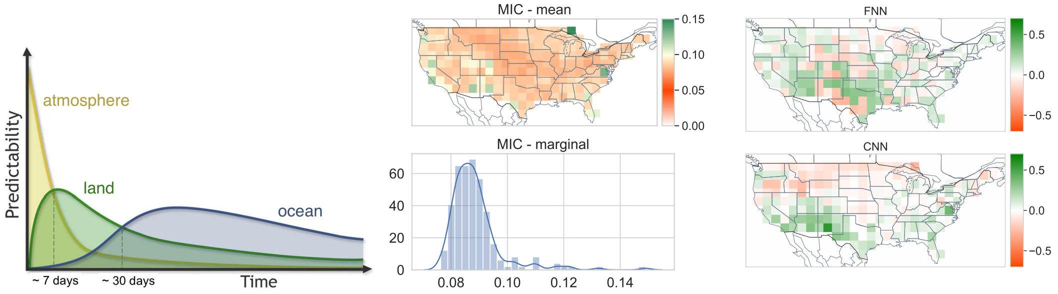
Part III
Advance Sub-seasonal Climate Forecasting with Machine Learning

Chapter 4

Sub-Seasonal Climate Forecasting via Machine Learning: Challenges, Analysis, and Advances

4.1 Introduction

Over the past few decades, major advances have been made in weather forecasts on time scales of days to about a week (Lorenc, 1986; National Research Council, 2010; Simmons and Hollingsworth, 2002). Similarly, major advances have been made in seasonal forecasts on time scales of 2-9 months (Barnston et al., 2012). However, making high-quality forecasts of key climate variables such as temperature and precipitation on sub-seasonal time scales, defined as the time range between 2-8 weeks, has long been a gap in operational forecasting (National Academies of Sciences, Engineering, and Medicine, 2016). Skillful climate forecasts at sub-seasonal time scales would be of immense societal value, and would have an impact in a wide variety of domains including agricultural productivity, water resource management, and emergency planning for extreme weather events, etc. (Klemm and McPherson, 2017; Pomeroy et al., 2002). The importance of sub-seasonal climate forecasting (SSF) has been discussed in great detail in two recent high profile reports from the National Academy of Sciences (National Academies of Sciences, Engineering, and Medicine, 2016; National Research Council, 2010). Despite the scientific, societal, and financial importance of SSF, progress on the problem has been limited (Braman et al., 2013; de Perez and Mason, 2014), partly because it has attracted less attention compared to weather and seasonal climate prediction. Also, SSF is arguably more difficult compared to weather or



(a) Sources of Predictability

(b) MIC

(c) Results of FNN and CNN

Figure 4.1: (a) Sources of predictability at different forecast time scales. Atmosphere is most predictive on weather time scales, whereas for SSF, land and ocean are considered important sources of predictability (Uccellini and Jacobs, 2018). (b) Maximum information coefficient (MIC) (Reshef et al., 2011) between residualized temperatures of week 3 & 4 and week -2 & -1. Small MICs (≤ 0.1) at a majority of locations indicate little information shared between the most recent date and the forecasting target. (c) Predictive skills of Fully connected Neural Networks (FNN) and Convolutional Neural Networks (CNN), in terms of temporal cosine similarity (see definition in Section 4.5), for temperature prediction over 2017-2018. FNN and CNN do not perform well, as the cosine similarities for most locations are either negative (red) or close to zero (white).

seasonal forecasting due to limited predictive information from land and ocean, and virtually no predictability from the atmosphere (Uccellini and Jacobs, 2018), which forms the basis of numerical weather prediction (Simmons and Hollingsworth, 2002) (Figure 4.1(a)).

There exists great potential to advance sub-seasonal prediction using Machine Learning (ML) techniques. Due in large part to this potential promise, a recently concluded real-time forecasting competition called the Sub-Seasonal Climate Forecast Rodeo was sponsored by the Bureau of Reclamation in partnership with NOAA, USGS, and the U.S. Army Corps of Engineers (Hwang et al., 2019; Raff et al., 2017). However, a direct application of standard black-box ML approaches to SSF can run into challenges due to the high-dimensionality and strong spatial correlation of the raw data from atmosphere, ocean, and land, e.g., Figure 4.1(c) shows that popular approaches such as Fully connected Neural Networks (FNN) and Convolutional Neural Networks (CNN) do not perform so well when directly applied to the raw data. One reason is that sub-seasonal forecasting does not lie in the big data regime: about 40 years of reliable data exists for all climate variables, with each day corresponding to one data point, which totals less than 20,000 data points. Additionally, different seasons may have different predictive relations, and many climate variables have strong temporal correlations at daily time scales, further reducing the effective data size. Therefore, it is worth carefully and systematically investigating the capability of ML approaches including Deep Learning (DL) while keeping in mind the high-dimensionality, spatial-temporal correlations, and limited observational data available for SSF. Our main contributions are as follows:

- We illustrate that, with the limited predictability at sub-seasonal time scale and the unique nature of climate data, i.e., strong spatial-temporal correlation, high-dimensionality, and limited amount of high-quality observational data, SSF imposes a great challenge for ML despite the recent advances in various domains.
- We perform a comprehensive empirical study on 10 ML approaches to SSF over the contiguous U.S. and show that suitable ML models, e.g., XGBoost, to some extent, capture predictability at sub-seasonal time scales and outperform existing approaches in climate science, such as climatology, i.e., the 30-year average at a given location and time. Notably, DL models are only able to match the best results after careful selection of architecture.
- We analyze and explore various aspects, e.g., feature representation and model architecture, which shed light on potential directions to improve the quality of sub-seasonal forecasts. Further, an analysis of feature importance suggests that ocean and land covariates are more useful than atmospheric covariates, which is consistent with Figure 4.1(a).
- We construct an SSF dataset covering the contiguous U.S. and including climate variables from the atmosphere, ocean, and land. We release the dataset and a flexible code base for data extraction, preprocessing, and SSF model training and evaluation.

Organization of the chapter. We discuss related work in Section 4.2. In Section 4.3, we describe the SSF problem tackled in this chapter and demonstrate its difficulties. In Section 4.4, we outline the investigated ML approaches. The details of experimental setup and results are provided in Section 4.5 and Section 6, and we conclude in Section 4.7.

4.2 Related Work

Although statistical models were used for weather prediction before the 1970s (Frederik Nebeker, 1995), since the 1980s weather forecasting has been carried out using mainly physics-based dynamical models (Barnston et al., 2012). More recently, there has been a surge of application for ML approaches to both short-term weather forecasting (Cofino et al., 2002; Grover et al., 2015; Radhika and Shashi, 2009), and longer-term climate prediction (Badr et al., 2014; Cohen et al., 2019). However, little attention has been paid on forecasting on sub-seasonal time scale (Vitart et al., 2012). Recently, ML techniques have made great strides in statistical prediction in many fields, so it is natural to investigate whether it can advance sub-seasonal climate prediction. In particular, many advances have occurred in developing prediction models using spatiotemporal climate data (Goncalves et al., 2017; Hwang et al., 2019; Steinhaeuser et al., 2011b), e.g., predicting land temperature using oceanic data (DelSole and Banerjee, 2017; He et al., 2019).

Since SSF can be formulated as a sequential modeling problem (Sutskever et al., 2014; Venugopalan et al., 2015), bringing the core strength of DL-based sequential modeling, a thriving research area, has great potential for a transformation in climate forecasting (Ham et al., 2019; Reichstein et al., 2019; Schneider et al., 2017). In the past decade, recurrent neural network (RNN) (Funahashi and Nakamura, 1993) and long short-term memory (LSTM) models (Gers et al., 2000) have become popular sequential models and have been successfully applied in language modeling and other seq-to-seq tasks (Sundermeyer et al., 2012). Starting from (Srivastava et al., 2015; Sutskever et al., 2014), the encoder-decoder structure with RNN or LSTM has become one of the most competitive algorithms for sequence transduction. Variants of such models that incorporate mechanisms like convolution (Shi et al., 2017; Xingjian et al., 2015) or attention mechanisms (Bahdanau et al., 2015) have achieved remarkable breakthroughs for audio synthesis, word-level language modeling, and machine translation (Vaswani et al., 2017).

SSF is an extremely important but largely understudied problem and ML is just starting to get used in this area. Within ML, (Hwang et al., 2019) are the first to specifically focus on SSF over western U.S. and released their benchmark dataset. In this work, we *expand* the spatial forecasting range to cover the entire contiguous U.S. and *extend* the set of predictors by including climate variables considered as important sources of predictability on sub-seasonal time scale (Uccellini and Jacobs, 2018), such as soil moisture, Niño and NAO indices.

4.3 Sub-seasonal Climate Forecasting

Problem statement. In this chapter, we focus on building temperature forecasting models at the forecast horizon of 15-28 days ahead, i.e., the target variable is the residualized average temperature of week 3 & 4. The geographic region of interest is the contiguous U.S. (latitudes 25N-49N and longitudes 76W-133W) at a 2° by 2° resolution (197 grid points).

Table 4.1: Description of climate variables and their data sources.

Type	Climate variable	Description	Unit	Spatial coverage	Data Source
Spatiotemporal	tmp2m	Daily average temperature at 2 meters	C°	Contiguous U.S.	CPC Global Daily Temperature (Fan and Van den Dool, 2008)
	sm	Monthly Soil moisture	mm		CPC Soil Moisture (Fan and van den Dool, 2004)
	sst	Daily sea surface temperature	C°	North Pacific & Atlantic Ocean	Optimum Interpolation SST (OISST) (Reynolds et al., 2007)
	rhum	Daily relative humidity near the surface (sigma level 0.995)	%	Contiguous U.S. and North Pacific & Atlantic Ocean	Atmospheric Research Reanalysi Dataset (Kalnay et al., 1996b)
	slp	Daily pressure at sea level	Pa		
	hgt10 & hgt500	Daily geopotential height at 10mb and 500mb	m		
Temporal	MEI	Bimonthly multivariate ENSO index	NA	NA	NOAA ESRL MEI.v2 (Zimmerman et al., 2016)
	Niño 1+2, 3, 3.4, 4	Weekly Oceanic Niño Index (ONI)			NOAA National Weather Service, CPC (Reynolds et al., 2007)
	NAO	Daily North Atlantic Oscillation index			NOAA National Weather Service, CPC (Van den Dool et al., 2000)
	MJO phase & amplitude	Madden-Julian Oscillation index			Australian Government BoM (Wheeler and Hendon, 2004)

Our covariates consist of climate variables, such as sea surface temperature, soil moisture, geopotential height, etc., indicating the status of land, ocean, and atmosphere. Table 4.1 provides a detailed description.

Difficulty of the problem. To illustrate the challenge of SSF, we measure the statistical dependence between the residualized average temperature of week -2 & -1 (1-14 days in the past) and week 3 & 4 (15-28 days in the “future”) at each grid point by maximum information coefficient (MIC) (Reshef et al., 2011), an information theory-based measure of the linear or non-linear association between two variables. The values of MIC range between 0 and 1, and a small MIC value close to 0 indicates a weak dependence. To assess statistical significance, we apply moving block bootstrap (Kunsch, 1989) to time series of 2-week average temperature at each grid point from 1986 to 2018, with the block size of 365 days. The top panel in Figure 4.1(b) illustrates the average MIC from 100 bootstrap over the contiguous U.S., and the marginal distribution of all locations is shown at bottom. Small MIC values (≤ 0.1), which indicate little predictive information shared between the most recent data and the forecasting target, to some extent, demonstrate how difficult SSF is.

From an ML perspective, applying black-box DL approaches naively to SSF is less likely to work due to the limited number of samples, and the high-dimensional and spatial-temporally correlated features. Figure 4.1(c) shows the performance of two vanilla DL models: FNN with ReLU activation function and CNN, in terms of the (temporal) cosine similarity between the prediction and the ground truth at each location over 2017-2018. For most locations, their cosine similarities are either negative or close to zero. Besides, as we illustrate in the sequel, we explore about 10 ML models for the problem, and most do not even get positive relative R^2, indicating that they perform no better than the long term average (details are presented in Appendix B.1). Such results further demonstrate the difficulty of the problem.

4.4 Methods

Notation. Let t denote a date and g denote a location. The target variable at time t and location g is the residualized average temperature over weeks 3 & 4 (from $t + 15$ to $t + 28$), denoted as $y_{g,t}$. For a given location g, $\mathbf{y}_{g,T}$ represents the target variable over a time range T. Similarly, $\mathbf{y}_{G,t}$ denotes the target variable over all G locations at time t. $X_t \in \mathbb{R}^p$ denotes the p-dimensional covariates at time t.

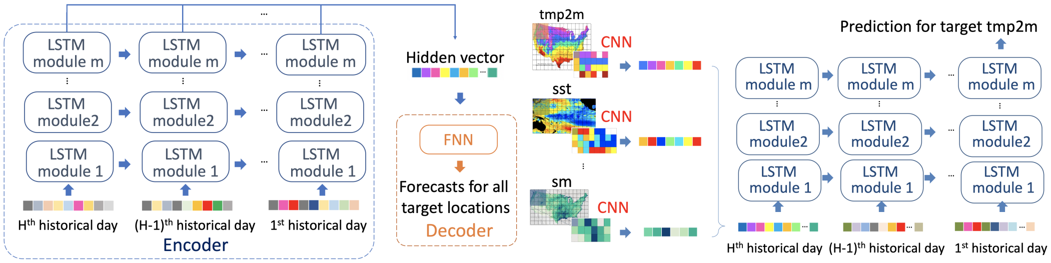
Non-DL models. We explore the following non-DL models.

- **MultiLLR** (Hwang et al., 2019). MultiLLR introduces a multitask feature selection algorithm to remove the irrelevant predictors and integrates the remaining predictors linearly. For a location g and a target date t^*, its coefficient β_g is estimated by

$$\hat{\beta}_g = \text{argmin}_\beta \sum_{t \in \mathcal{D}} w_{g,t}(y_{g,t} - \beta^T X_t)^2 \ , \tag{4.1}$$

  where $\mathcal{D}$ is the temporal span around the target date's day of the year and $w_{g,t}$ is the corresponding weight. In (Hwang et al., 2019), an equal data point weighting ($w_{g,t} = 1$) has been employed.

- **AutoKNN** (Hwang et al., 2019). An auto-regression model with weighted temporally local samples, where the auto-regression lags are selected via a multitask k-nearest neighbor criterion. The method only takes historical measurements of the target variables as input. The nearest neighbors of each target date are selected based on an average of spatial cosine similarity computed over a history of $M = 60$ days, starting one year prior to a target date t^* (lag $l = 365$). More precisely, the similarity between the target date t^* and a date t in the corresponding training set is formulated



(a) Encoder (LSTM)-Decoder (FNN)

(b) CNN-LSTM

Figure 4.2: Architectures of the designed DL models. (a) Encoder (LSTM)-Decoder (FNN) includes a few LSTM layers as the Encoder, and two fully connected layers as the Decoder. (b) CNN-LSTM consists of a few convolutional layers followed by an LSTM.

as $\text{sim}_t = \frac{1}{M}\sum_{m=0}^{M-1}\cos(\mathbf{y}_{G,t-l-m}, \mathbf{y}_{G,t^*-l-m})$, where $\cos(\mathbf{y}_{G,t_1}, \mathbf{y}_{G,t_2})$ computes the (spatial) cosine similarity (see formal definition in Section 4.5), evaluated over G locations, between two given dates t_1 and t_2.

- **Multitask Lasso** (Jalali et al., 2013; Tibshirani, 1996). It assumes $\mathbf{y}_{G,t} = X_t\Theta^* + \epsilon$, where $\epsilon \in \mathbb{R}^G$ is a Gaussian noise vector and $\Theta^* \in \mathbb{R}^{p\times G}$ is the coefficient matrix for all locations. With n samples, Θ^* is estimated by $\hat{\Theta}_n = \text{argmin}_{\Theta\in\mathbb{R}^{p\times G}} \frac{1}{2n}\|Y - X\Theta\|_2^2 + \lambda_n\|\Theta\|_{2,1}$ with $X \in \mathbb{R}^{n\times p}$ and $Y \in \mathbb{R}^{n\times G}$. λ_n is a penalty parameter and the corresponding penalty term is computed as $||\Theta||_{2,1} = \sum_i(\sum_j \Theta_{ij}^2)^{1/2}$.

- **Gradient boosted trees (XGBoost)** (Chen and Guestrin, 2016; Friedman, 2001). A functional gradient boosting algorithm using regression tree as its weak learner. The algorithm starts with one weak learner and iteratively adds new weak learners to approximate functional gradients. The final ensemble model is constructed by a weighted summation of all weak learners.

- **State-of-the-art climate baselines.** We consider two baselines from climate science perspective, both are Least Square (LS) linear regression models (Weisberg, 2005). The first model uses covariates based on climate indices, such as NAO and Niño indices, which are widely used to monitor ocean conditions. The covariate of the second model is the most recent available data point from target variable, i.e, residualized temperature of week -2 & -1, with which the model, also known as *damped persistence* (Van den Dool, 2007) in climate science, is essentially a first-order autoregressive model.

DL models. We design two DL models, namely Encoder (LSTM)-Decoder (FNN) and

CNN-LSTM, specifically adapting to SSF. The objective function is to minimize the mean squared error (ℓ_2 loss) among all dates and locations.

- **Encoder (LSTM)-Decoder (FNN).** Inspired by Autoencoder widely used in sequential modeling (Sutskever et al., 2014), we design the Encoder (LSTM)-Decoder (FNN) model, of which the architecture is shown in Figure 4.2(a). Input of the model is features extracted spatially from covariates using unsupervised methods like Principal Component Analysis (PCA). The temporal components of covariates are handled by feeding features of each historical date into an LSTM Encoder recurrently. Then, the output of each date from LSTM is sent jointly to a two-layer FNN network with ReLU activation function. The output of the FNN Decoder is the predicted residualized temperature of week 3 & 4 over all target locations.
- **CNN-LSTM.** The proposed CNN-LSTM model directly learns the representations from the spatiotemporal data using convolutional layers. Shown in Figure 4.2(b), CNN extracts features for each climate variable at all historical dates separately. Then, the extracted features from the same date are collected and fed into an LSTM model recurrently. The temperature prediction for all target locations is done by an FNN layer taking the output of the LSTM's last layer from the latest input.

4.5 Data and Experimental Setup

Data description. Climate agencies across the world maintain multiple datasets with different formats and resolutions. We construct the SSF dataset by collecting climate variables (Table 4.1) from a diverse collection of data sources and converting them into a consistent format. In particular, temporal variables, e.g., Niño indices, are interpolated to a daily resolution, and spatiotemporal variables are interpolated to a spatial resolution of 0.5° by 0.5°.

Preprocessing. Spatiotemporal climate variables are normalized by z-scoring at each location and each date using the mean and standard deviation of the corresponding day of the year over 1986-2016. Temporal covariates, e.g., Niño indices, are directly used without normalization. CNN and CNN-LSTM take the temporal and normalized spatiotemporal variables as input. Models other than CNN based models, e.g., XGBoost and Multitask Lasso, can not directly use spatiotemporal covariates due to the extremely high dimensionality of such covariates. In those cases, we extract the top 10 principal components (PCs) of each spatiotemporal covariate, based on PC loadings from 1986 to 2016 (for details, refer

Table 4.2: Comparison of spatial cosine similarity of tmp2m forecasting for test sets over 2017-2018. XGBoost and Encoder (LSTM)-Decoder (FNN) have the best performance. Models achieve better performance using temporally global set compared to temporally local set.

Model	Mean(se)	Median (se)	0.25 quantile (se)	0.75 quantile (se)
Temporally Global Dataset				
XGBoost - one day	**0.3044(0.03)**	**0.3447(0.05)**	**0.0252(0.05)**	**0.5905(0.04)**
Lasso - one day	0.2499(0.04)	0.2554(0.06)	-0.0224(0.05)	0.5604(0.06)
Encoder (LSTM)-Decoder (FNN)	**0.2616 (0.04)**	**0.2995 (0.07)**	**-0.0719 (0.06)**	**0.6310 (0.05)**
FNN	0.0792(0.01)	0.0920(0.02)	0.0085(0.02)	0.1655(0.02)
CNN	0.1688(0.04)	0.2324(0.06)	-0.0662(0.06)	0.4768(0.04)
CNN-LSTM	0.1743(0.04)	0.2867(0.06)	-0.1225(0.07)	0.5148(0.04)
LS with NAO & Niño	**0.2415(0.03)**	**0.3169(0.04)**	**0.0454(0.05)**	**0.4624(0.03)**
Damped persistence	0.2009(0.04)	0.2310(0.06)	-0.0884(0.06)	0.5335(0.05))
MultiLLR	0.0684 (0.03)	0.1046 (0.05)	-0.1764 (0.06)	0.3156 (0.04)
AutoKNN	0.1457 (0.03)	0.1744 (0.05)	-0.1018 (0.06)	0.4000 (0.04)
Temporally Local Dataset				
XGBoost - one day	0.1965(0.04)	0.2345(0.05)	-0.0636(0.06)	0.5178(0.05)
Lasso - one day	0.1631(0.04)	0.2087(0.06)	-0.1178(0.05)	0.5059(0.05)
Encoder (LSTM)-Decoder (FNN)	0.1277 (0.04)	0.1272 (0.06)	-0.1558 (0.06)	0.4971 (0.06)

to Appendix B.2), and normalize PCs by z-scoring at each day of the year. For all models, the target variable is the residualized 2m temperature over the contiguous U.S. via the same normalization as spatiotemporal climate variables.

Feature set construction. We combine the PCs of spatiotemporal covariates with temporal covariates into a sequential feature set, which consists not only covariates of the target date, but also covariates of the 7^{th}, 14^{th}, and 28^{th} day prior to the target date, as well as the day of the year of the target date in the past 2 years and both the historical past and future dates around the day of the year of the target date in the past 2 years (see Appendix B.2 for a detailed example).

Evaluation pipeline. Predictive models are created independently for each month in 2017 and 2018. To mimic a live system, we generate 105 test dates during 2017-2018, one for each week, and group them into 24 test sets by their month of the year. Given a test set, our evaluation pipeline consists of two parts: (1) "5-fold" training-validation pairs for hyperparameter tuning, based on a "sliding-window" strategy designed for time-series data. Each validation set consists of the data from the same month of the year as the test set, and we create 5 such sets from dates in the past 5 years (2012 - 2016). Their corresponding training sets contain 10 years of data before each validation set; (2) the training set, including 30-year data in the past. To assure no overlap between the training and test set, we enforce the training set to end 28 days before the first date in the test set. We share more explanations, including a pictorial example, in Appendix B.2.

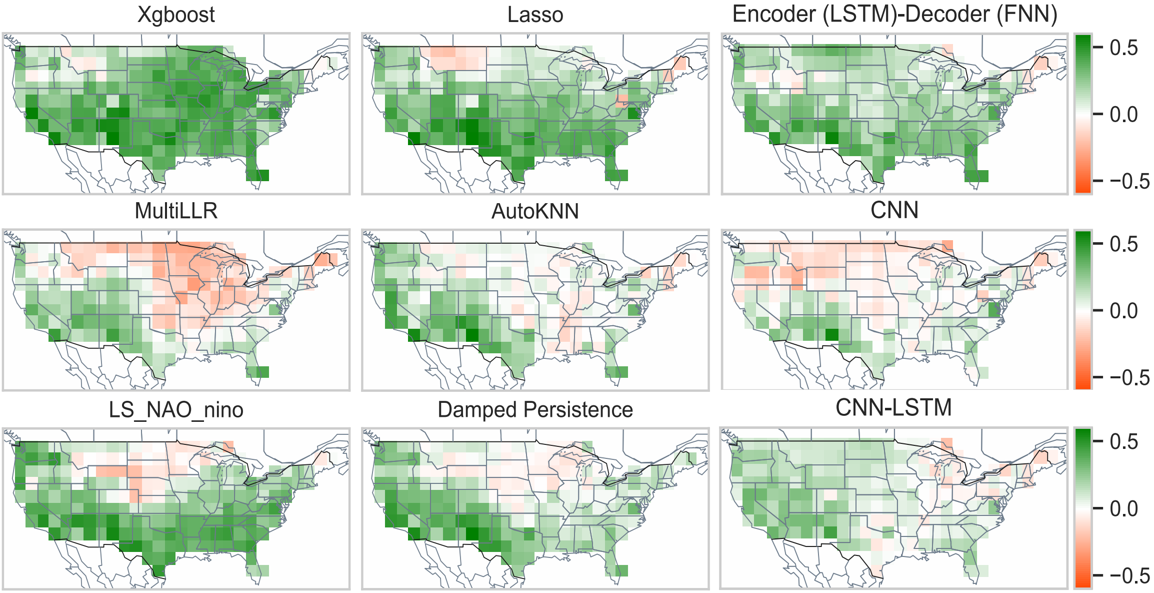


Figure 4.3: Temporal cosine similarity over the contiguous U.S. of ML models for temperature prediction over 2017-2018. Large positive values (green) closer to 1 indicate better predictive skills. Overall, XGBoost and Encoder (LSTM)-Decoder (FNN) perform the best. Qualitatively, coastal and south regions are easier to predict than inland regions (e.g., Midwest).

Evaluation metrics. Forecasts are evaluated by cosine similarity, the only metrics used in the Sub-Seasonal Climate Forecast Rodeo (Raff et al., 2017). The cosine similarity between $\hat{\mathbf{y}}$, a vector of predicted values, and $\mathbf{y}^*$, the corresponding ground truth, is computed as $\cos(\hat{\mathbf{y}}, \mathbf{y}^*) = \frac{\langle \hat{\mathbf{y}}, \mathbf{y}^* \rangle}{\|\hat{\mathbf{y}}\|_2 \|\mathbf{y}^*\|_2}$, where $\langle \hat{\mathbf{y}}, \mathbf{y}^* \rangle$ denotes the inner product between the two vectors. Then, the spatial cosine similarity is defined as $\cos(\hat{\mathbf{y}}_{G,t}, \mathbf{y}^*_{G,t})$, measuring the prediction skill at a date t. The temporal cosine similarity, assessing the prediction skill at a location g, is defined as $\cos(\hat{\mathbf{y}}_{g,T}, \mathbf{y}^*_{g,T})$.

4.6 Experimental Results

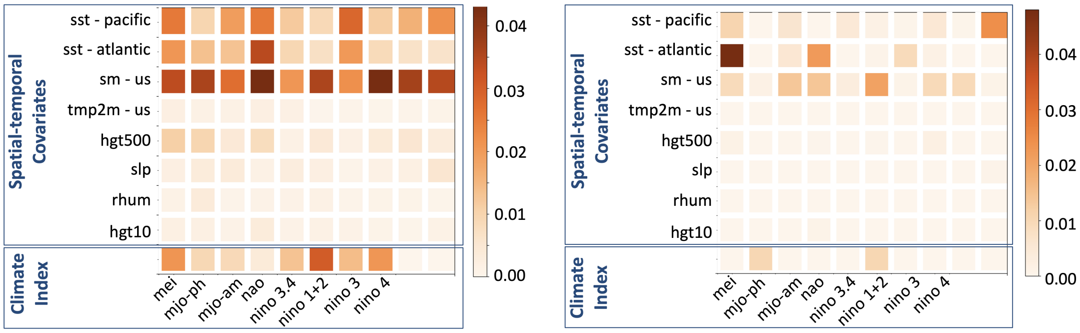
We compare the predictive skills of 10 ML models on SSF. In addition, we discuss a few aspects that impact the ML models the most, as well as the evolution of our DL models.

4.6.1 Results of All Methods

Temporal results. Table 4.2 lists the mean, the median, the 0.25 quantile, the 0.75 quantile, and their corresponding standard errors of spatial cosine similarity of all methods. Additional results based on relative R^2 can be found in Appendix B.3. XGBoost, Encoder (LSTM)+Decoder (FNN) and Lasso accomplish higher predictive skills than other presented methods and can outperform climatology and two climate baseline models, i.e., LS with NAO & Niño, and damped persistence. Overall, XGBoost achieves the highest predictive skill in terms of both the mean and the median, demonstrating its predictive power. Surprisingly, linear regression with a proper feature set has good predictive performance. Even though DL models are not the obvious winner, with careful architectural selections, they still show some encouraging results.

Spatial results. Figure 4.3 shows the temporal cosine similarity of all methods evaluated on test sets described in Section 4.5. Among all methods, XGBoost and the Encoder (LSTM)-Decoder (FNN) achieve the overall best performance, regarding the number of locations with positive temporal cosine similarity. Qualitatively, coastal and south regions in general are easier to predict compared to inland regions (e.g., Midwest), which might be explained by the influence of the slow-moving component, i.e., Pacific and Atlantic Ocean. Such component exhibits inertia or memory, in which anomalous condition can take relatively long period of time to decay. However, each model has its own favorable and disadvantageous regions. For example, XGBoost and Lasso do poorly in Montana, Wyoming, and Idaho, while Encoder (LSTM)-Decoder (FNN) performs much better on those regions. The observations naturally imply that the ensemble of multiple models is a promising future direction.

Comparison with the state-of-the-art methods. MultiLLR and AutoKNN are two state-of-the-art methods designed for SSF on western U.S. (Hwang et al., 2019). Both methods have shown good forecasting performance on the original target region. However, over the inland region (Midwest), Northeast, and South region, the methods do not perform so well (Figure 4.3). To be fair, even though a similar set of climate variables have been used in our work compared to the original paper (Hwang et al., 2019), how we prepossess the data and construct the feature set are slightly different. Such differences may lead to relatively poor performance for these two methods, especially for MultiLLR. A detailed comparison over western U.S. and on SubseasonalRodeo dataset (Hwang et al., 2019) can be found in Appendix B.3.



(a) XGBoost

(b) Lasso

Figure 4.4: SHAP values computed from (a) XGBoost and (b) Lasso. Darker color means a covariate is of the higher importance. The first 8 rows contains the top 10 principal components extracted from 8 spatiotemporal covariates respectively, and the last row includes all temporal indices. Land covariate, e.g., soil moisture and ocean covariates, e.g., sst and some climate indices, are considered more important.

4.6.2 Analysis and Exploration

We analyze and explore several important aspects that could influence the performance of ML models.

Temporally "local" vs. "global" dataset. Our current training set consists of all calendar months over the past 30 years, which we refer to as the temporally "global" dataset. Another way to construct the training set is to only consider calendar months within a temporal neighborhood of the test date. For instance, to make forecasts of June 2017, the training set can contains dates in June (from earlier years), and months that are close to June, e.g., April, May, July, and August, over the past 30 years only. Such a construction accounts for the seasonal dependence of predictive relations, for example summer predictions are not trained with winter data. We name such dataset as a temporally "local" dataset. A comparison between the "global" and "local" datasets has been listed in Table 4.2 where a significant drop in cosine similarity can be noticed when using "local" dataset for all of our best predictive models, including XGBoost, Lasso, and Encoder (LSTM)-Decoder (FNN). We suspect such performance drop from "global" to "local" dataset may come from the reduction in the number of effective samples.

Feature importance. We study which covariates are important, considered by ML models, based on their SHAP (SHapley Additive exPlanations) values (Lipovetsky and Conklin, 2001; Lundberg and Lee, 2017). SHAP values illustrate how much each feature contributes to the forecasts. Therefore, features with large absolute SHAP values are important. Figure 4.4 shows the mean of absolute SHAP values for each covariate over 24 models (one per month in 2017-2018), computed from (a) XGBoost and (b) Lasso. Among all covariates, soil moisture (3^{rd} row from the top) is the variable that has been constantly considered as important covariates by both models. Another set of important covariates is the family of Niño indices. An LS model using those indices alone as predictors performs fairly well (Table 4.2). Besides, sst of both Pacific and Atlantic also stand out. Such observations indicate that ML models pick up ocean-based covariates, some land-based covariates, and almost entirely ignore the atmosphere-related covariates, which are well aligned with domain knowledge (Delsole and Tippett, 2017; Uccellini and Jacobs, 2018).

The influence of feature sequence length. To adapt the usage of LSTM, we construct a sequential feature set, which consists not only the target date, but also 17 other dates preceding the target date. However, other ML models, e.g., XGBoost and Lasso, which are not designed to handle sequential data, experience a drastic performance drop when we include more information from the past. More precisely, by including covariates from the full historical sequence, the performance of XGBoost drops approximately 50% compared to when using covariates from the most recent date only. A possible explanation is that, as we increase the feature sequence length, such model weights covariates from different dates exactly the same without considering temporal relationship, thus irrelevant historical information might mislead the model. In Appendix B.3, we compares results obtained from various sequence lengths.

4.6.3 What Happened with DL Models?

While applying black-box DL models naively does not work well for SSF, the improvement (Table 4.2), as we evolve from FNN to CNN-LSTM, and finally to Encoder (LSTM)-Decoder (FNN), demonstrates how the network architecture plays an important role. Below we focus on discussing feature representation and the architecture design for sequence modeling. More discussions are included in Appendix B.3.

Feature representation: CNN vs. PCA. Since SSF can be considered as a spatiotemporal prediction problem, to handle the spatial aspect, CNN can be applied as a “supervised” way for learning feature representation by viewing each climate covariate as a map. CNN, while doing convolution using a small kernel, mainly focus on spatially localized regions. However, the global dependency among climate variables restricts the effectiveness of CNN kernels on feature extraction, which explains the limited predictive skill of CNN shown in Table 4.2 and Figure 4.3. Meanwhile, PCA, termed Empirical Orthogonal Functions

(EOF) (Von Storch and Zwiers, 2001) in climate science, is a commonly used “unsupervised” feature representation method, which focuses on low-rank modeling of spatial covariance structure revealing spatial connection. By using PCs, we are including spatial and temporal information about the dominant components of variability in each spatiotemporal covariate. Our results (Table 4.2) illustrate that PCA-based models have higher predictive skills than CNN-based models, verifying that PCA is an adequate technique for feature extraction in SSF.

Sequential modeling: Encoder-Decoder. With features extracted by PCA, we formulate SSF as a sequential modeling problem (Sutskever et al., 2014), where the input is the covariates sequence described in Section 4.5, and the output is the target variable. Due to the immense success in sequential modeling (Srivastava et al., 2015), the standard Encoder-Decoder, where both Encoder and Decoder are LSTM (Hochreiter and Schmidhuber, 1997), is the first architecture to investigate. Unfortunately, the model does not perform well and suffers from over-fitting, possibly caused by overly complex architecture. To reduce the model complexity, we replace the LSTM Decoder with an FNN Decoder which takes only the last step of the output sequence from the Encoder. Such change leads to an immediate boost of predictive performance. However, the input of the FNN Decoder mainly contains information encoded from the latest day in the input sequence and can only embed limited amount of historical information owing to the recurrent architecture of LSTM. To further improve the performance, we adjust the connection between Encoder and Decoder, such that FNN Decoder takes every step of the output sequence from LSTM Encoder, which makes a better use of historical information. Eventually, such architecture achieves the best performance among all investigated Encoder-Decoder variants (see comparisons in Appendix B.3).

4.7 Conclusions

In this chapter, we investigate the potential to advance sub-seasonal climate forecasting, a challenging and understudied problem, using ML techniques. SSF is typically a high-dimensional problem on strongly spatiotemporal correlated climate data with limited number of samples. We conduct a comprehensive study of 10 ML models, including DL models, on the SSF dataset, which is constructed for SSF over the contiguous U.S. Empirical results show the gradient boosting model XGBoost, the DL model Encoder (LSTM)-Decoder (FNN), and the linear model Lasso manage to outperform forecasts based on climatology, damped persistence and climate indices. Besides, our analysis and exploration provide insight on several essential aspects to improve the SSF performance, and show that ML models are capable of picking the climate variables from important sources of predictability on sub-seasonal time scale. We release the SSF dataset and code base publicly, which will hopefully reduce the barrier to work on SSF for the broader ML community.

Chapter 5

Learning and Dynamical Models for Sub-seasonal Climate Forecasting: Comparison and Collaboration

5.1 Introduction

Nowadays, weather forecasts are routinely available out to a few days, and seasonal forecasts are routinely available out to a few months. These forecasts are based largely on dynamical models that solve partial differential equations (PDEs) derived from the laws of physics. On the other hand, sub-seasonal forecasting (SSF), which refers to the prediction of key climate variables, e.g., temperature and precipitation on 2-week to 2-month time scales, are not yet routined. Nevertheless, two high-profile reports from the National Academy of Sciences (NAS) discuss the immense societal values of SSF (National Academies of Sciences, Engineering, and Medicine, 2016; National Research Council, 2010). In particular, skillful SSF in the western contiguous United States would allow for better hydrology and water resource management, and emergency planning for extreme events such as droughts and wildfires (White et al., 2017). Currently, sub-seasonal forecasts based on dynamical models are available weekly through the Subseasonal Experiment (SubX) project (Pegion et al., 2019), but the full utility of these for operational forecasting still remains to be determined.

SSF is challenging for a variety of reasons. First, atmospheric weather is chaotic, meaning that forecasts are sensitive to small differences in the initial condition (Lorenz, 1963).

Furthermore, the target time window is beyond the 2-week period for which individual weather systems can be predicted, but shorter than the 3-month period over which weather variability can be filtered by time averaging (National Research Council, 2010). Also, from a physical point of view, the predictability on sub-seasonal time scales depends on correctly modeling the atmosphere, ocean, and land, including their interactions and couplings as well as the memory effects of land and ocean. In addition to these physical complexities, SSF poses unconventional time series prediction problems. Given a training set $\{x_{1:t}, y_{1:t}\}$, where y denotes the target response variable, e.g., land temperature, and x denotes suitable covariates, temporal models typically focus on predicting y_{t+1} or maybe $y_{t+1:t+\tau_s}$ for a small τ_s (a few days or less). Instead, SSF is about predicting $y_{t+T:t+T+\tau_l}$ for large $T \gg \tau_s$, e.g., weather prediction one month ahead ($T = 31$ days). The long temporal range relative to the weather predictability time, along with the nonlinear dynamics and complex interactions, makes SSF challenging.

For climate forecasting, one standard baseline for comparing forecasts is the so-called climatology (Trewin et al., 2007). Typically, the climatology is defined as the 30-year average temperature/precipitation for each calendar day at each geographic location. Note that the climatology is merely the historical average without using any initial condition information. Despite its simplicity, the climatology provides a competitive benchmark for SSF. For instance, in the last Forecast Rodeo (NIDIS, 2019), a SSF competition sponsored by the U.S. Bureau of Reclamation and the NOAA/National Integrated Drought Information System (USBR and NOAA, 2019), real-time predictions of sub-seasonal temperature and precipitation were submitted by different groups but about half of the submitted forecasts could not beat climatology. Therefore, for any other possibly more advanced SSF model, the first order of business is to do better than this strong climatological baseline. In recent years, progress has been made in developing ML models (He et al., 2021a; Hwang et al., 2019; Srinivasan et al., 2021; Weyn et al., 2021) which have shown great promise for SSF, including outperforming the climatology.

In this chapter, we consider two new directions: first, comparing and contrasting ML models for SSF with an arguably stronger baseline provided by physics-based dynamical models for SSF, and second, exploring enhancing the ML models by using forecasts from such dynamical models. For the comparison, earlier literature have done such comparisons with certain statistical approaches and have illustrated dynamical models to have better forecasting ability (Barnston et al., 2012). Instead, we do the comparison with a suite of modern ML methods, including non-parametric AutoKNN (Hwang et al., 2019), multitask Lasso (Jalali et al., 2013; Tibshirani, 1996), gradient boosted trees (Chen and Guestrin, 2016; Friedman, 2001), and deep encoder-decoder networks (He et al., 2021a), and illustrate that on average ML models outperform dynamical models on SSF. With considerably more details, our empirical analysis demonstrates key properties of ML-based vs. dynamical model based predictions. In particular, ML models are more conservative in their forecasts whereas dynamical models are more aggressive, so that when dynamical models are wrong, they can be wrong by a large amount; on the flip side, when dynamical models are correct, they can be more accurate than ML models. Further, we illustrate that ML models make most of their bad predictions during extreme events, e.g., unusual cold waves in North America for which there is not enough training data. More practically, these results suggest that a separate ML model for extreme events will help improve the aggregate performance. The second direction is using physics-based dynamical model forecasts as covariates in the ML models. We show that using dynamical model forecasts as inputs improves the ML model forecasts, and the improvements are statistically significant.

We briefly emphasize the contributions of our work. We are *not* proposing another new algorithm which improves performance on an existing task using benchmark datasets, such as MNIST (Lecun et al., 1998) and ImageNet (Deng et al., 2009). We are enabling a new application area for data mining based on one of the most challenging and societally important scientific problems in the context of climate forecasting. We are reporting promising results using ML models, reporting detailed and nuanced empirical analysis acknowledging the strengths of both ML and dynamical models, and illustrating gains by using dynamical model forecasts as covariates in the ML models. We also suggest ways of further improving the ML models, e.g., by separately modeling extreme events. Finally, the dataset constructed for this work, dynamical model predictions, and code for the ML models will be made available to enable the data mining community improve on the current results. We also hope that the SSF dataset will become a standard benchmark like MNIST(Lecun et al., 1998) or ImageNet(Deng et al., 2009), and accelerate advances on SSF.

5.2 Related Work

Dynamical models and S2S forecasting. Nowadays, weather predictions rely heavily on ensemble forecasts from physics-based dynamical models (Barnston et al., 2012). On sub-seasonal to seasonal (S2S) time scales, forecasts have shown limited predictive skill compared to the climatology (Vitart, 2004,1; Weigel et al., 2008). However, successful S2S predictions can be performed for certain regions and seasons (Delsole and Tippett, 2017; Li and Robertson, 2015), as well as certain climate states (Mariotti et al., 2020). In order to better understand the conditions that lead to enhanced predictability and to improve sub-seasonal forecasts, projects such as S2S (Vitart et al., 2017) and SubX (Pegion et al., 2019) have been established. These coordinated multi-model efforts act both to fulfill the growing needs of real world applications and to enrich our understanding of S2S prediction and predictability.

ML on weather and S2S forecasting. Recently, increasing efforts have been made to tackle complex problems in climate science using machine learning. Such applications aim to advance weather forecast skill using deep learning methods (Dueben and Bauer, 2018; Ham et al., 2019; Liu et al., 2016; Scher and Messori, 2019). Despite early studies that show dynamical models outperform statistical models for ENSO seasonal forecasts (Barnston et al., 2012), recent advances in machine learning, especially the development of deep learning, are making the performance of ML models more competitive with dynamical models for both weather (Dueben and Bauer, 2018; Grover et al., 2015; Shi et al., 2017)and seasonal (Stevens et al., 2021) prediction.

In particular, machine learning models have started to be used to improve forecast skills for predictions of temperature, precipitation, and other climate variables on sub-seasonal time scales (He et al., 2021a; Hwang et al., 2019; Srinivasan et al., 2021; Weyn et al., 2021). Some successful ML approaches for S2S forecasting include (Hwang et al., 2019) and (He et al., 2021a), where both works show increased predictive skill for ML models compared to climatic baselines, e.g., climatology and damped persistence. Such advances from ML models are particularly relevant and valuable because dynamical models have limited predictive skills at sub-seasonal time scales (Uccellini and Jacobs, 2018).

5.3 Sub-seasonal Climate Forecasting

5.3.1 Problem Statement

For our comparison, we follow the Forecast Rodeo competition, which is a SSF competition sponsored by the U.S. Bureau of Reclamation and the NOAA/National Integrated Drought Information System (USBR and NOAA, 2019). The details of this competition were motivated by the needs of water managers, where skillful information on weather and climate conditions could enhance the efficient utilization of water resources to reduce the impact of hydrological variations. Broadly speaking, the competition is focused on SSF of temperature and precipitation over the western contiguous U.S. In this study, we focus on forecasting temperature over days 15 - 28, i.e., predicting average temperatures over week

3 & 4 ahead of time, over the region bounded by latitudes 25N-50N and longitudes 93W-125W at 1° by 1° spatial resolution with 508 grid points. The specific temporal range of interest and temporal resolutions are determined by each SubX model and its initialization frequency (see Table 5.1).

5.3.2 Ground Truth Dataset

The ground truth dataset is constructed from NOAA's Climate Prediction Center (CPC) Global Gridded Temperature dataset (Fan and Van den Dool, 2008), which contains observations from the Global Telecommunication System (GTS) gridded using the Shepard Algorithm (Shepard, 1968). Commonly applied for forecast verification by NOAA/CPC (Fan and Van den Dool, 2008), the CPC dataset provides daily max and min 2m temperatures (tmp2m) at 0.5 ° by 0.5 ° spatial resolution from Jan 1, 1979 to present.

To obtain the ground truth temperature anomalies for weeks 3 &4, we preprocess the data as follows: (1) daily 2m temperature at each grid point is taken as the average of daily max and min tmp2m, (2) all missing values are imputed by averaging the daily tmp2m of its spatial/temporal neighbors, (3) the tmp2m at 0.5 ° × 0.5° resolution are linearly interpolated to a 1 ° × 1° grid, (4) the daily tmp2m anomalies are computed by subtracting the climatology from the observed daily tmp2m, and (5) the forecasting target at each date and grid point is the average of tmp2m anomalies at day 15 to 28. The climatology used in step (4) is the smoothed long-term average of tmp2m over 1990 - 2016 for each month-day combination and grid point. Specifically, for a given grid point, we compute the long-term average over 1990 - 2016, one for each month-day combination. Then the 365 values are smoothed using moving average with a window size of 31 days.

5.3.3 Evaluation Metrics

We consider three metrics to evaluate the predictive performance of each forecasting model. Let $\mathbf{y}^* \in \mathbb{R}^n$ denotes a vector of ground truth observation and $\hat{\mathbf{y}} \in \mathbb{R}^n$ the corresponding predicted value.

(Uncentered) Anomaly Correlation Coefficient (ACC) (Wilks, 2011) is defined as

$$\mathrm{ACC} = \frac{\langle \hat{\mathbf{y}}, \mathbf{y}^* \rangle}{\|\hat{\mathbf{y}}\|_2 \|\mathbf{y}^*\|_2} \ , \tag{5.1}$$

where $\langle \hat{\mathbf{y}}, \mathbf{y}^* \rangle$ denotes the inner product between the two vectors. Uncentered anomaly correlation is the only metric used in the Sub-Seasonal Climate Forecast Rodeo Competition

Table 5.1: Summary of GMAO-GEOS and NCEP-CFSv2.

SubX Model	Ensemble Members	Initialization Interval	Hindcast Range	Forecast Range
GMAO-GEOS	4	5	01-01-1999 to 12-31-2015	07-25-2017 to 06-30-2020
NCEP-CFSv2	4	1	01-01-1999 to 12-31-2015	07-01-2017 to 03-15-2020

(Hwang et al., 2019; Raff et al., 2017).

Relative R^2 is defined as

$$\begin{aligned} \text{relative } R^2 &= 1 - \text{Relative MSE} \qquad (5.2) \\ &= 1 - \frac{\sum_{i=1}^{n}(\mathbf{y}_i^* - \hat{\mathbf{y}}_i)^2}{\sum_{i=1}^{n}(\mathbf{y}_i^* - \bar{\mathbf{y}}_{\text{train}})^2} , \qquad (5.3) \end{aligned}$$

where $\bar{\mathbf{y}}_{\text{train}}$ is the long-term average of tmp2m at each date and grid point in the training set. Relative R^2 represents the relative skill against the best constant predictor, i.e., $\bar{\mathbf{y}}_{\text{train}}$. A model which achieves a positive relative R^2 is, at least, able to predict the sign of y^* accurately and outperforms the climatology.

Root Mean Square Error (RMSE) is defined as

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\mathbf{y}_i^* - \hat{\mathbf{y}}_i)^2}{n}} . \qquad (5.4)$$

where $\mathbf{y}_i^*$ and $\hat{\mathbf{y}}_i$ are the i-th element in $\mathbf{y}$ and $\hat{\mathbf{y}}$ respectively.

Denote the ground truth temperature anomalies as $Y^* \in \mathbb{R}^{T \times G}$, where T is the number of dates and G is the number of grid points. The spatial predictive skill for a given date t can be evaluated on $\mathbf{y}_t^* = Y^*[t, :]$, the t-th row in Y^*, where $\mathbf{y}_t^* \in \mathbb{R}^G$ is the ground truth for all grid points at date t with the corresponding forecasts $\hat{\mathbf{y}}_t$. The temporal predictive skill for a grid point g can be evaluated on $\mathbf{y}_g^* = Y^*[:, g]$, the g-th column in Y^*, similar to time series prediction evaluation.

5.4 Subseasonal Experiment (SubX) Project

The Subseasonal Experiment (SubX) is a project that provides sub-seasonal forecasts from multiple global forecast models. Data are publicly available through the International Research Institute for Climate and Society (IRI) Data Library at Columbia University. A detailed description of the SubX project and the contributing models can be found in (Pegion et al., 2019). The SubX project has two predictive periods: hindcast and forecast. A hindcast period represents the time when a dynamic model re-forecasts historical events, which can help climate scientists develop and test new models to improve forecasting. Hindcasts in the SubX project occurred during January 1999 to December 2015. In contrast, a forecast period has real-time predictions generated from dynamic models. The real-time forecast period in the SubX project starts from July 2017. In this work we evaluate the predictive skills of the SubX models over their forecast periods.

In this chapter, we focus on two SubX models, NCEP-Climate Forecast System version 2 (CFSv2) (Saha et al., 2014) and NASA-Global Modeling and Assimilation (GMAO) version 2 of the Goddard Earth Observing System (GEOS) model (Reichle and Liu, 2014). NCEP-CFSv2 is a coupled atmosphere–ocean–land–ice model and is the operational seasonal prediction model currently used by the U.S. Climate Prediction Center. The NCEP-CFSv2 forecasts are initialized daily and include four ensemble members. In order to ensure our results are not unique to a single dynamical model, we also analyze output from the GMAO-GEOS. The GMAO-GEOS is also fully coupled atmosphere–ocean–land–sea ice model, with forecasts initialized every five days and includes four ensemble members. We selected the GMAO-GEOS model for comparison because it has an initialization frequency (every 5 days) that is closer that of the NOAA-CSFv2 (daily). Other SubX models were initialized less frequently. Besides, our code base and the ground truth dataset are fairly flexible, which can easily be extended to evaluate other SubX models.

Further information of the two SubX models are presented in Table 5.1. For both SubX models, the average of four ensemble members' outputs are taken as the forecasts. All forecasts include daily values for 45 days beyond the initialization date. The weeks 3 & 4 outlooks are computed by averaging the forecasts 15 to 28 days beyond each initialization date and subtracting the corresponding climatology computed from the model's hindcast period.

5.5 Machine Learning-based SSF Modeling

Notation. Let $Y \in \mathbb{R}^{T \times G}$ denote the targeted weeks 3 & 4 temperature anomalies over T dates and G grid points. $\mathbf{y}_t$ is the t-th row in Y, denoting the temperature anomalies over all grid points G at date t. $X \in \mathbb{R}^{T \times p}$ denotes the p-dimension covariates for T dates. $X_t \in \mathbb{R}^p$ (the t-th row in X) is the covariate at date t.

5.5.1 Machine Learning Models

In this chapter, we focus on state-of-the-art machine learning models which have been shown to work effectively for sub-seasonal climate forecasting (He et al., 2021a; Hwang et al., 2019).

AutoKNN (Hwang et al., 2019). An auto-regressive model only uses features from historical temperature anomalies, which selects lagged measurements with a multitask k-nearest neighbor criterion. For a given date t, the algorithm chooses the temperature anomalies from 20 historical dates with the highest similarity and 29 days, 58 days, and 1 year prior to t as features. Specifically, the similarity between two dates t_1 and t_2 is defined as $\text{sim}_{(t_1,t_2)} = \frac{1}{M}\sum_{m=0}^{M-1}\cos(\mathbf{y}_{t_1-l-m}, \mathbf{y}_{t_2-l-m})$, where $\cos(\mathbf{y}_{t_1-l-m}, \mathbf{y}_{t_2-l-m})$ is the cosine similarity between the temperature anomalies at $l-m$ days before t_1 and t_2. Following the settings in (Hwang et al., 2019), we use $M = 60$, the length of the considered historical sequences prior to each date, with the lag $l = 365$. At each grid point, we fit a weighted local linear regression model, where the weight is one over the variance of the temperature anomalies at the corresponding date.

Multitask Lasso (Jalali et al., 2013; Tibshirani, 1996). A multitask regularized linear regression model. By assuming $\mathbf{y}_t = X_t\Theta^* + \epsilon$, where $\epsilon \in \mathbb{R}^G$ is a Gaussian noise and $\Theta^* \in \mathbb{R}^{p\times G}$ is the coefficient matrix for all locations, the parameter Θ^* is estimated by

$$\hat{\Theta} = \text{argmin}_{\Theta\in\mathbb{R}^{p\times G}} \frac{1}{2T}\|Y - X\Theta\|_2^2 + \lambda\|\Theta\|_{2,1} \tag{5.5}$$

with $||\Theta||_{2,1} = \sum_i(\sum_j \Theta_{ij}^2)^{1/2}$ and λ being the penalty parameter.

Gradient boosted trees (XGBoost) (Chen and Guestrin, 2016; Friedman, 2001). A functional gradient boosting algorithm, of which the weak learners are regression trees. The algorithm combines multiple weak learners into a stronger learner in an iterative manner. At each iteration, a new weak learner is created to correct the previous prediction and optimize the loss function along with regularization. We build one XGBoost model for each location, and the hyper-parameters are selected jointly based on the performance over all locations.

Encoder-FNN (He et al., 2021a) A deep learning model designed for SSF over the contiguous U.S. The architecture is shown in Figure 4.2(a). The model input is a historical sequence of the features shared by all locations and is fed into an LSTM encoder recurrently. The output of each step in the sequence are combined and jointly sent to the decoder, which is a two-layer fully-connected neural network with ReLU activation. The outputs of the decoder are the predicted tmp2m anomalies over all grid points. Note that, besides standard hyper-parameters like layer size, number of layers, and dropout rate, the length of the sequence

Table 5.2: Description of climate variables and their data sources.

Type	Climate variable	Description	Data Source
Spatiotemporal	tmp2m	Daily temperature at 2 meters	CPC Global Daily Temperature (Fan and Van den Dool, 2008)
	sm	Monthly soil moisture	CPC Soil Moisture (Fan and van den Dool, 2004)
	sst	Daily sea surface temperature	Optimum Interpolation SST (OISST) (Reynolds et al., 2007)
	rhum	Daily relative humidity near the surface (sigma level 0.995)	Atmospheric Research Reanalysis Dataset (Kalnay et al., 1996b)
	slp	Daily pressure at sea level	
	hgt10 & hgt500	Daily geopotential height	
Temporal	MEI.v2	Bimonthly multivariate ENSO index	NOAA ESRL MEI.v2 (Zhang et al., 2019)
	MJO phase & amplitude	Madden-Julian Oscillation index	Australian Government BoM (Wheeler and Hendon, 2004)
	Niño 1+2, 3, 3.4, 4	Weekly Niño index	NOAA National Weather Service, CPC (Reynolds et al., 2007)
	NAO	Daily North Atlantic Oscillation index	NOAA National Weather Service, CPC (Van den Dool et al., 2000)
	SSW	Sudden Stratospheric Warming index (The zonal mean winds at 60N at 10hPa)	Modern-Era Retrospective Analysis for Research and Applications v2 (Gelaro et al., 2017)

is also a hyper-parameter. The final forecast is the average of 20 independent runs.

5.5.2 Covariates for ML Models

The feature set for the ML models contains the following climate variables. Spatially over the contiguous U.S. we consider (1) 2m temperature (tmp2m), which is also the source data for the ground truth dataset; (2) soil moisture, which influences temperature and precipitation through its impact on surface fluxes of heat and moisture (Koster et al., 2011); and (3) four climate variables - geopotential height (ght) at 10mb and 500mb, sea level pressure (slp) and relative humidity (rhum) - from the reanalysis dataset, which capture variations in the northern hemisphere polar vortex and persistent variations in the large-scale atmospheric circulation. We also obtain sea surface temperature (sst) over the Pacific Ocean, from latitudes 20S to 65N and longitudes 120E to 90W, and the Atlantic Ocean, from latitudes 20S to 50N and longitudes 20W to 90W. Variations in sst have been linked to enhanced sub-seasonal predictability over the U.S.(DelSole et al., 2017).

In addition, we include nine climate indices that describe the state of the climate system or are related to different climate phenomena, such as El Niño/Southern Oscillation (ENSO). Multivariate ENSO index (MEI.v2) and Niño indices are included for monitoring El Niño and La Niña events (DelSole et al., 2017; Stan et al., 2017). The amplitude and phase of Madden-Julian Oscillation are considered since the MJO has dramatic impacts in the mid-latitudes and is a strong contributor to various extreme events in the U.S. (Waliser, 2005). North Atlantic Oscillation index (NAO) is considered since variations in the NAO drive changes in temperature and precipitation over the U.S. and western Europe (Stan et al., 2017). Sudden Stratospheric Warming index (SSW) is included to capture the variations in the strength of the polar vortex, which are associated with extreme cold air outbreaks in mid-latitude U.S., Europe, and Asia (Butler et al., 2015).

5.5.3 Data Preprocessing

For all ML models except AutoKNN, we consider two types of climate variables, namely spatiotemporal and temporal climate variables. For each spatiotemporal variable, we flatten the values at all grid points for each date and compute the top 10 principal components (PCs) as features. For example, if $X^{\mathrm{sm}} \in \mathbb{R}^{T\times G}$ denotes the soil moisture for T dates in training set (1990-2016) and all G spatial grid points over the contiguous U.S., we compute the PC loadings using X^{sm} and extract the top 10 PCs to get the feature matrix $X^{\mathrm{sm}}_{\mathrm{pc}} \in \mathbb{R}^{T\times 10}$. The extracted PCs are then normalized by z-scoring for each month-day combination separately. The temporal variables and the PC-based features of all spatiotemporal climate variables jointly form the feature set for each date. For XGBoost and Lasso, the covariates are the feature values two weeks lagging from the forecasting period. For example, if the forecasting period is Jan, 15 - Jan, 28 in 2019, the covariates are the features on Jan 1, 2019. For Encoder FNN, the features of a historical sequence are treated as the model input for each date. The historical sequence is constructed similarly to the features of Encoder FNN shown in Figure B.2 (He et al., 2021a). AutoKNN takes only the historical tmp2m anomalies as the covariate.

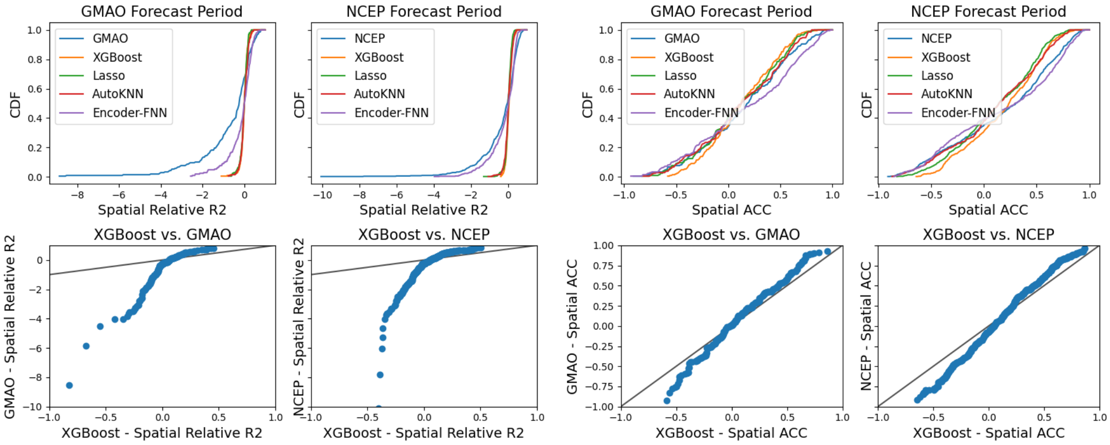
5.5.4 Experimental Setup

Since the relationships between the covariates and target variables vary at different times of the year, test sets are created for each month from July 2017 to Jun 2020 and separate predictive models are trained accordingly. Since an individual ML model is built for each month of the year, the best hyper-parameters of each type of ML models are selected on a monthly basis. To do so, for each month of the year, we construct five validation sets containing data from the same month between 2012 and 2016, and the corresponding training sets consist of 10 years of data prior to each validation set. The best hyper-parameters are determined by the average performance over the five validations sets. We thus have twelve sets of the best hyper-parameters corresponding to each month of the year. Once the best hyper-parameters are selected, we use 28 years of data prior to a given test set to train the corresponding ML forecasting model.

5.6 Experimental Results

In this section, we compare the predictive skill of the four ML models and the two SubX models on the forecast period from 2017 to 2020. A comprehensive analysis is conducted for the experimental results, which reveals possible directions for further improvement of the ML models for SSF. Besides, we explore the potential of advancing SSF by combining the ML models and the SubX forecasts.

5.6.1 Forecast Period Evaluation



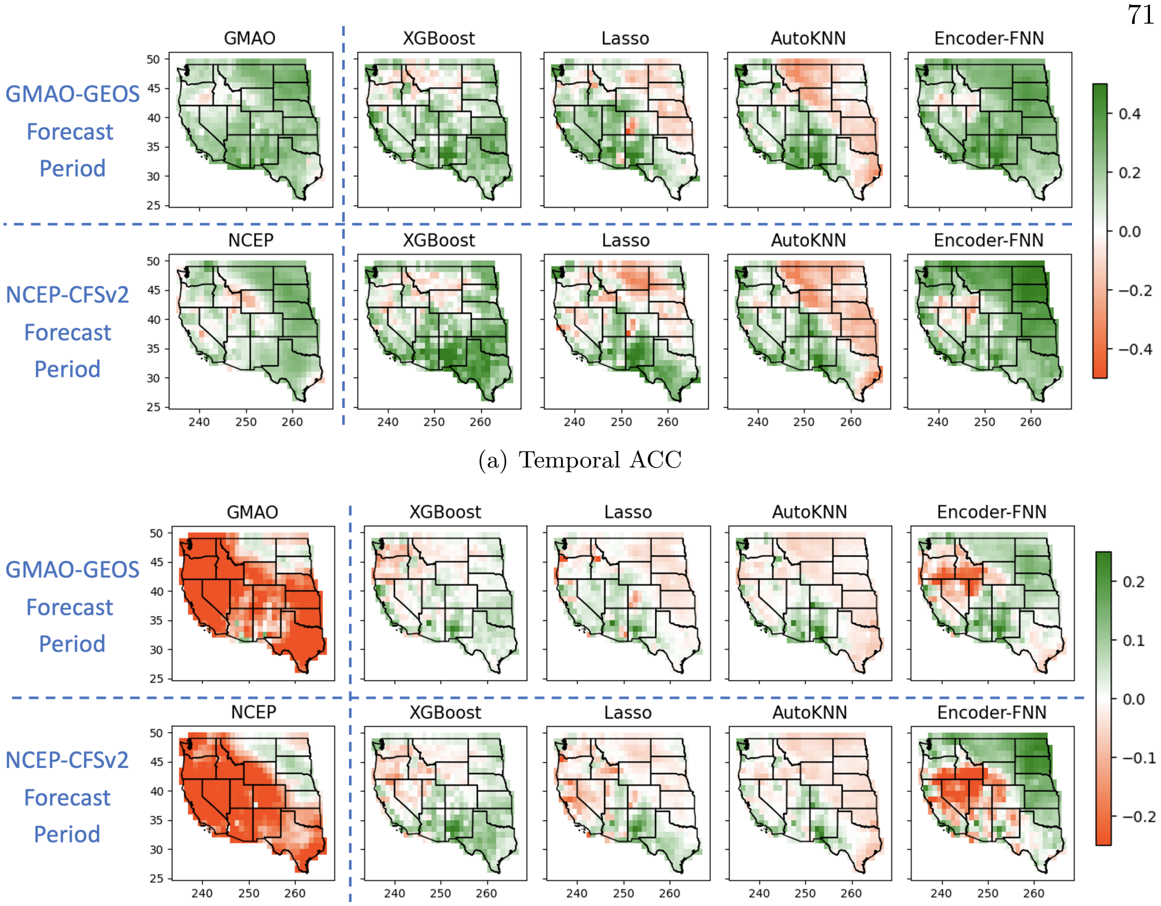
(a) The plots for spatial relative R^2.

(b) The plots for spatial ACC.

Figure 5.1: (a) The empirical cumulative distribution function (cdf) of spatial relative R^2 (top) of all methods and the quantile-quantile (QQ) plot of relative R^2 (bottom) between XGBoost and GMAO-GEOS (left) or NCEP-CFSv2 (right). XGBoost, Lasso and AutoKNN all have spatial relative R^2 close to or above 0, while the SubX models and Encoder-FNN have relative R^2 much smaller than -1. (b) The cdf and QQ plot of spatial ACC. Despite the similarity of the cdf curves, the ML models (yellow, green, and red) are in general below the blue curve (the SubX model) when the spatial ACCs are negative, which indicates that the ML models are less likely to have extremely negative predictive skills compared to the SubX models.

Since the GMAO-GEOS and NCEP-CFSv2 models have different forecast periods and temporal resolutions, all results are evaluated at their respective forecast periods and resolutions. We first present the empirical cumulative distribution function (cdf) of spatial relative R^2 for all methods over the forecast periods of GMAO-GEOS and NCEP-CFSv2 in Figure 5.1(a). It is shown that ML models such as XGBoost, Lasso, and AutoKNN are capable of generating forecasts with positive or smaller negative relative R^2, while the SubX models and the Encoder-FNN model commonly stay in the negative relative R^2 zone. On the other hand, considering the positive side of the cdf plot, the SubX model and Encoder-FNN are able to achieve relative R^2 close to 1 in some cases, whereas the cdf of other ML models reach 1 when the relative R^2 are comparatively small. The quantile-quantile (QQ) plot of spatial relative R^2 in Figure 5.1(a) shows that the relative R^2 can be much smaller than -1, indicating the SubX models can make predictions with a large deviation from the ground truth. A similar set of plots for spatial ACC is presented in Figure 5.1(b). Despite the similarities in the cdf across models, a closer inspection shows the cdf of the ML models (yellow, green, and red curves) are generally below the cdf of the SubX models (blue curve) for spatial ACC between [-1, 0]. The QQ plot of the spatial ACC between XGBoost and SubX models supports the observation, where all the points are below the diagonal line when the spatial ACC of XGBoost is between [-1, 0]. For the positive side of the spatial ACC for XGBoost, most points are close to or slightly above the diagonal line. To summarize, at a given date, the SubX models are more likely to have spatial ACC close to the extreme values (-1 or +1), while ML models, such as XGBoost, are more conservative and are able to avoid extreme negative ACC.

The temporal ACC and temporal relative R^2 over the western U.S. are illustrated in Figure 5.2(a) and (b) respectively. Similar to spatial results, the SubX models achieve positive temporal ACC for most spatial locations while performing poorly with respect to temporal relative R^2. Among all ML models, XGBoost and Encoder-FNN are the best two considering temporal predictive skills and substantially outperform the SubX models for most spatial locations, especially compared to NCEP-CFSv2. Spatially, the central area, including the states of North Dakota, South Dakota, Montana, Wyoming, Kansas, and Oklahoma, are the areas where the temperature fluctuations are more drastic compared to the coastal states. Therefore, linear model like Lasso and non-parametric model like AutoKNN tend to perform worse in such regions, while more complicated nonlinear models like XGBoost and Encoder-FNN perform relatively better. Additionally, the SubX models have negative temporal relative R^2 and positive temporal ACC for the coastal area, which implies the SubX models may predict incorrect magnitudes despite their relatively accurate prediction of the temporal patterns.



(a) Temporal ACC

(b) Temporal Relative R^2

Figure 5.2: (a) Temporal ACC and (b) temporal relative R^2 of the SubX models (leftmost columns) and the ML models. For both metrics, values closer to 1 (green) indicate more accurate predictions. Overall the SubX models achieve positive temporal ACC for most spatial locations while performing poorly if considering temporal relative R^2. Among all the ML models, XGBoost and Encoder-FNN are the two best models regarding both predictive skills, and substantially outperform the SubX models over most spatial locations, especially compared to NCEP-CFSv2.

5.6.2 Machine Learning and Extreme Weather Events

Given that SSF is a challenging problem, it is natural to investigate under which circumstance(s) the ML models fail to provide accurate forecasts. The average spatial ACC of the XGBoost models and the SubX models for each month during the forecast periods are shown in Figure 5.3. For most months, XGBoost is either competitive or achieves higher spatial ACC compared to the SubX models. The exceptions occur in December 2018 and

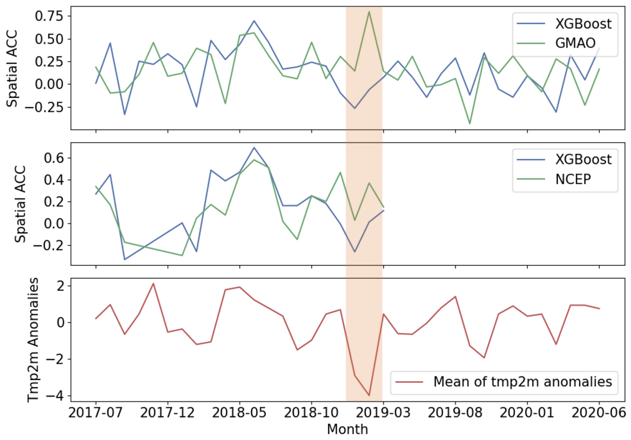


Figure 5.3: The monthly average spatial ACC of XGBoost and the SubX models (top 2) during the respective forecast periods and the mean of tmp2m anomalies over the western U.S. (bottom). Most of the time, XGBoost achieves competitive or even higher spatial ACC compared to the SubX models. The only exception, that both SubX models outperform XGBoost, happens from Dec. 2018 to Feb. 2019 (highlighted in orange) when a cold wave affected the U.S. leading to extreme low average tmp2m anomalies.

first two months of 2019, when the January–February 2019 North American cold wave impacted the United States. The cold wave brought the coldest temperatures in over 20 years to most locations (Wikipedia, 2019). The temperature anomalies reached -15°C and beyond in the *central U.S.* Extreme weather events are hard to predict since there is a lack of enough training data for such events. However, the dynamical models are reasonably successful in predicting the extreme cold temperatures, since they follow the physics. For example, the cold wave followed a sudden stratospheric warming event, which have been shown to increase predictability of these extreme events (Domeisen and Butler, 2020).

The value of spatial ACC is not affected by the scale of the response. Therefore, we analyze the predictive performance regarding RMSE for the ML models. We first separate all grid points in the western U.S. into five climatically consistent regions (Karl and Koss, 1984), i.e., northwest, west, west-north-central, southwest, and south (Figure 5.4). To represent the spatial variance of tmp2m anomalies at each forecasting date and each region, we approximately compute the standard deviation (std) of tmp2m anomalies at each date and each region as $\sqrt{\sum_{i=1}^{n_r} \frac{y_i^2}{n_r}}$, where n_r is the number of grid points for a given region at one date. As shown in Figure 5.5(a), the RMSE from all four ML models at a given date and region is strongly correlated to the std of tmp2m anomalies, which implies the dates and regions with high variance are difficult to predict. Figure 5.5(b) illustrates the average of tmp2m (with sign, unlike Figure 5.5(a)) for each date and region versus the

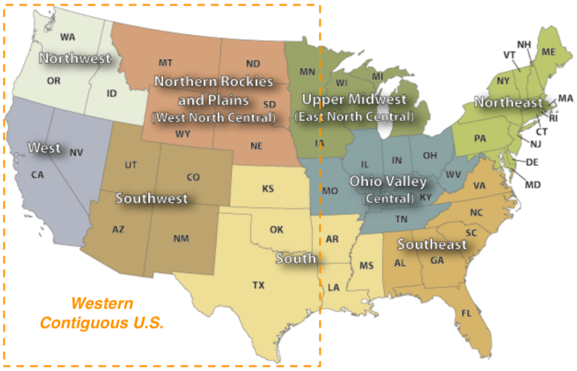


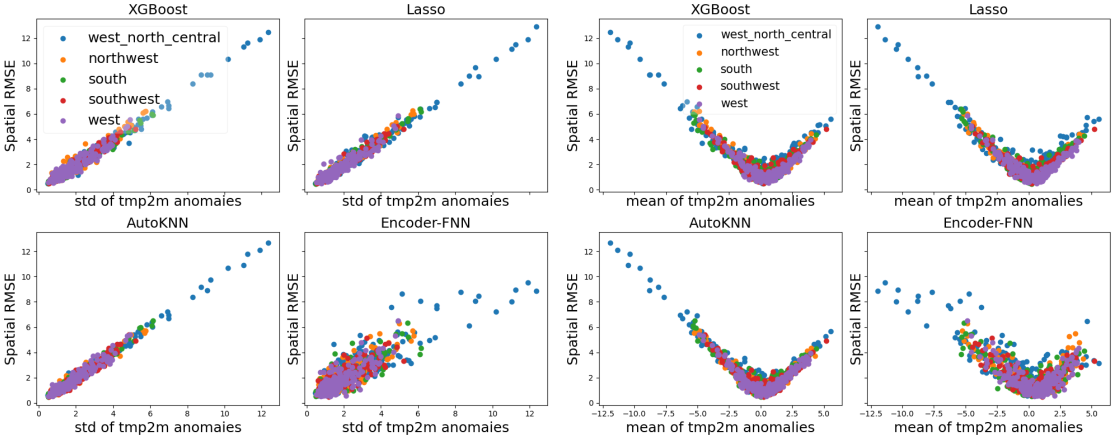
Figure 5.4: The nine climatically consistent regions identified by National Centers for Environmental Information scientists(NOAA - National Centers for Environmental Information, 2021; Karl and Koss, 1984). The western contiguous U.S. (orange rectangular) covers five regions with 20 states.

predictive RMSE, which further demonstrates that extreme events are the samples with negative bias and large variance during the forecast period. Besides, the distribution of different regions in Figure 5.5 implies that the spatial variance is, in general, lower for coastal regions compared to inland regions. For instance, west-north-central region can experience extremely cold winter temperatures when the polar jet stream sinks down into the mid-latitudes and brings with it the coldest polar air.

This analysis illustrates the difficulty of modeling extreme weather events using a single ML model, not only because of the inadequate samples, but also due to the intense temperature fluctuations caused by such events. Therefore, it is necessary to utilize separate modeling techniques for weather extremes or regions with drastic fluctuations in tmp2m anomalies, to achieve more accurate predictions. Ideally, if weather extremes can be detected ahead of time, we can choose not to trust the ML forecasts for a certain time period and turn to the forecasting models specifically designed for extreme conditions.

5.6.3 Enhancing ML Models with SubX Forecasts

To demonstrate the strengths and limitations of the SubX and the ML model forecasts, we present forecasts of two days as anecdotal evidence in Figure 5.6. The first example (Figure 5.6(a)) shows that, on Mar. 12, 2018, both GMAO-GEOS and XGBoost have successfully reproduced the spatial pattern of the ground truth. As a result, GMAO-GEOS and XGBoost obtain good spatial ACC. However, the predicted scale from GMAO-GEOS is much larger than XGBoost and is closer to the scale of the ground truth. The second example is the forecasting results on Jan 6, 2020, when the SubX forecasts fail badly. As



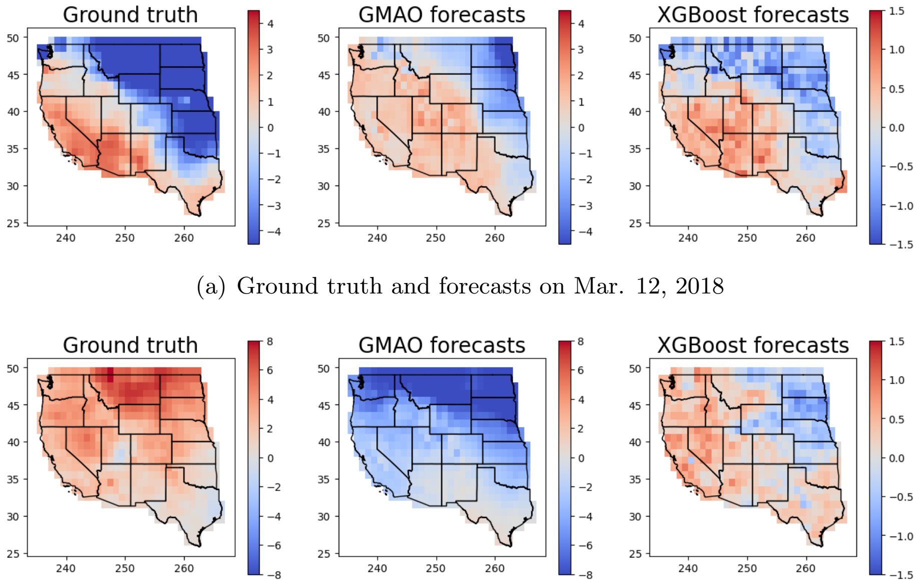
(a) Spatial RMSE vs. std of tmp2m anomalies

(b) Spatial RMSE vs. average tmp2m anomalies

Figure 5.5: Spatial RMSE versus (a) the standard deviation of tmp2m anomalies over dates and regions and (b) the average tmp2m anomalies over the dates and regions. The high spatial RMSE appears for samples having large standard deviation or extreme negative average of tmp2m anomalies, which indicates that the west-north-central region are hard to predict.

shown in Figure 5.6(b), while the ground truth is that all the locations over the western U.S. have positive tmp2m anomalies with the largest values around 8°C, GMAO-GEOS predicts all negative tmp2m anomalies with the lowest values close to $-8°C$. Meanwhile, XGBoost partially predicts the correct spatial pattern but with conservative values in the range of $[-1.5°C, 1.5°C]$, which are much smaller than the magnitudes of the ground truth. These two examples demonstrate that the SubX models have certain advantage on matching the magnitude of the tmp2m anomalies, while the ML models are more conservative and provide predicted values with smaller amplitude. On the flip side, in situations where the SubX models does not predict the spatial pattern correctly, the forecasts can be wrong by a large amount.

Acknowledging the advantages of both types of models, we explore a suitable combination of the ML models and the SubX forecasts. More specifically, we investigate whether including SubX forecasts in the feature set of the ML models can enhance the predictive skill of the ML models. Since the hindcast periods of the SubX models are ∼10 years shorter than the temporal range of the training data for the ML models, and the temporal resolution of SubX models is also relatively lower, incorporating the SubX forecasts significantly reduces the sample size. To compare the performance fairly, we first train a ML model using the samples that are available during the hindcast periods and then compare



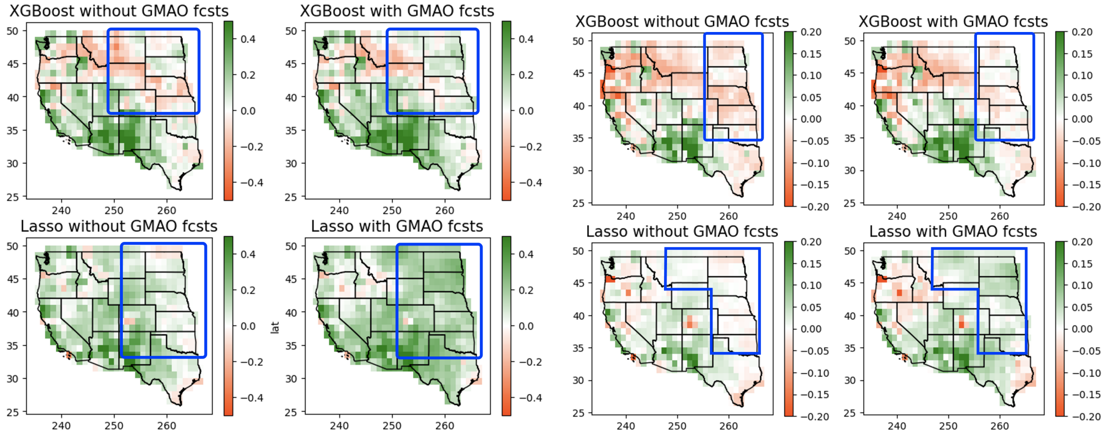
(a) Ground truth and forecasts on Mar. 12, 2018

(b) Ground truth and forecasts on Jan. 6, 2020

Figure 5.6: Comparison between the ground truth and forecasts made by GMAO-GEOS and XGBoost at two selected dates. (a) On March 12, 2018, both XGBoost and GMAO-GEOS successfully predict the spatial pattern of ground truth data (red in the southwest and blue in the northeast). However, predicted values from XGBoost are much smaller than GMAO-GEOS forecasts as XGBoost is more conservative on the magnitude. (b) On Jan 6, 2020, the ground truth has positive tmp2m anomalies (red) for most locations, while GMAO-GEOS mistakenly makes extreme negative forecasts (dark blue).

it with version of the ML model that uses SubX forecasts as features, this guarantees both models are trained with exactly the same sample size. Note, for Multitask Lasso, features are originally shared for all locations. To incorporate SubX forecasts, we have to build one Lasso model for each location but the hyper-parameter is jointly selected based on the performance for all locations.

Table 5.3 presents the mean and median and their standard error of the spatial ACC using XGBoost and Lasso, with and without the inclusion of SubX forecasts in the feature set. Temporal results are shown in Figure 5.7. Overall adding either GMAO-GEOS or NCEP-CFSv2 forecasts in the feature set leads to a significant enhancement of predictive skill. We conduct the sign test introduced in (DelSole and Tippett, 2016) to compare differences in forecast skills. Overall, comparison of ML model performance with and without SubX features yields p values much smaller than 0.01. The one exception is the spatial



(a) Temporal ACC

(b) Temporal relative R^2

Figure 5.7: The temporal ACC and relative R^2 of XGBoost and Lasso with and without GMAO forecasts as features.Including GMAO forecasts in the feature set evidently improves the forecasting performance, especially for the central U.S. (top right corner, marked by blue frames).

ACC for Lasso with and without GMAO forecasts, for which the p value is 0.21. Furthermore, as shown in Figure 5.7, the combination of the ML models and the SubX forecasts effectively converts some negative temporal ACC to positive and strengthens the forecasts originally achieving positive temporal ACC. The improvement is particularly outstanding for the west-north-central region, a region considered hard to predict. Similarly, regarding temporal relative R^2, both ML models obtain some improvements in the areas originally characterized by negative values. Especially for Lasso, it picks the central area where the GMAO-GEOS model performs well and obtains positive temporal relative R^2. These results highlight the potential to further increase predictive skill of the ML models by incorporating SubX forecasts. We anticipate that more hindcast data from SubX models would lead to notable improvement in predictive skills of the ML models.

5.7 Conclusions

In this chapter, we perform a rigorous evaluation and comparison between state-of-the-art machine learning models and two dynamical models from the SubX project, i.e., GMAO-GEOS and NCEP-CFSv2, for SSF in the western contiguous U.S. Experimental results demonstrate that, in general, the ML models can outperform the SubX models. However, the ML model forecasts usually are relatively conservative compared to the SubX forecasts

Table 5.3: The mean and median (standard error) of spatial ACC of XGBoost and Lasso with and without including the SubX forecasts in their feature set. Their spatial ACC have improved significantly when the SubX forecasts are included.

Features	without GMAO	with GMAO	without NCEP	with NCEP
XGBoost				
Mean	0.09 (0.02)	**0.13** (0.02)	0.15 (0.02)	**0.18** (0.02)
Median	0.12 (0.03)	**0.14** (0.04)	0.21 (0.03)	**0.23** (0.02)
Lasso				
Mean	0.12 (0.03)	**0.16** (0.03)	0.19 (0.01)	**0.23** (0.02)
Median	0.16 (0.04)	**0.18** (0.04)	0.21 (0.02)	**0.25** (0.02)

which, when correctly made, match the scale of the ground truth better. Acknowledging the strengths of both ML and dynamical models, we obtain significant improvements in predictive skill by including the SubX forecasts as a new feature of ML models, which illustrates the potential in generating skillful SSF by combining such two types of models. Further, we show that ML models make most of the bad forecasts during weather extremes, e.g., unusual cold waves, and suggest ways of further improving the ML models by separately modeling extreme events.

Chapter 6

Synthetic Climate Data Generation Using Generative Models

6.1 Introduction

As illustrated in the two previous chapters, sub-seasonal climate forecasting is a challenging problem for machine learning algorithms. One main reason is that SSF does not lie in the big data regime due to the limited availability of high-resolution climate data. Therefore, in this chapter, we focus on generating synthetic climate data using generative modeling.

In machine learning, generative modeling targets capturing the distribution of observed data $\mathbf{x}$ or the relationships between observed data $\mathbf{x}$ and some unobservable latent variables $\mathbf{z}$ (Ghahramani, 2015). Recently, due to the breakthroughs in deep learning, deep generative models, which seek a rich latent representation of data, have wide-ranging applications (An and Cho, 2015; Kingma et al., 2019; Kusner et al., 2017; Pu et al., 2016), as diverse as from image generation (Cai et al., 2019) to natural language processing (Bowman et al., 2016). The two main paradigms in generative modeling are Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) and Variational Autoencoders (VAEs) (Kingma and Welling, 2014; Kingma et al., 2019). GANs implicitly estimate the density via a stochastic procedure that generates data directly, whereas VAEs aim explicitly to approximate the posteriors for latent variables via maximizing the lower bound of the data log-likelihood.

The popularity of GANs has drastically increased in the past few years since they are able to produce images that are visually appealing and realistic (Creswell et al., 2018; Goodfellow et al., 2016). However, it is hard to train GANs due to the issues like non-convergence, mode collapse, and unbalance between the generator and discriminator (Arjovsky and Bottou, 2017). Recent studies (Arjovsky et al., 2017; Metz et al., 2016; Poole et al., 2016; Salimans et al., 2016) have shown various ways to stabilize the training of GANs. In particular, (Arjovsky et al., 2017) proposed an alternative with better theoretical properties than vanilla GANs, named Wasserstein GAN (WGAN), which leverages the Wasserstein-1 distance, rather than Jensen–Shannon (JS) divergence (used in vanilla GANS), between the model and target distributions. WGAN has the constraints that its discriminator must be a 1-Lipschitz function, which is enforced through weight clipping. Later, to prevent some undesired behaviors introduced by weight clipping in WGAN, Gulrajani et al. (2017) propose gradient penalty (WGAN-GP), which outperforms WGAN and achieves high-quality image generations on various benchmarking datasets (Karras et al., 2018).

On the other hand, VAEs use the lower bound of log-likelihood as the objective function, which targets modeling the underlying distribution of data. Unlike GANs, VAEs are relatively easier and more stable to train (Kingma, 2017), but vanilla VAEs tend to generate blurry images (Higgins et al., 2017; Hu et al., 2018). To improve VAEs, recent studies have devoted to conquer the statistical challenges, including formulating tighter bounds (Alemi et al., 2018; Burda et al., 2016; Masrani et al., 2019), specifying more flexible approximate posterior distributions (Cremer et al., 2018; Gregor et al., 2015; Kingma et al., 2016; Rezende and Mohamed, 2015; Vahdat et al., 2020), addressing the latent variable collapse problem (Bowman et al., 2015; Lucas et al., 2019; Razavi et al., 2018), and training VAEs with discrete latent variables (Rolfe, 2016; Tucker et al., 2017; Vahdat et al., 2018), etc. Besides, some recent work focuses on improving the interpretability of VAEs via learning disentangled representation (Chen et al., 2018; Higgins et al., 2017; Locatello et al., 2019) and generating high-quality images using deeper (hierarchical) architectures (Child, 2020; Maaløe et al., 2019; Vahdat and Kautz, 2020).

Inspired by the success of deep generative modeling in computer vision and natural language processing, some deep generative models have been introduced in weather and climate modeling as well. For example, deep generative models have been used for precipitation nowcasting (Ravuri et al., 2021), downscaling (Cheng et al., 2021), and wind speed prediction (Fanfarillo et al., 2021), etc. In this chapter, we investigate the performance of deep generative models for synthetic climate data generation. We propose a novel Vision Transformer-based variational autoencoder model (ViT-VAE) which combines the state-of-the-art computer vision model with VAE. In addition, we carefully compare the proposed model with another popular type of generative model, i.e., WGAN-GP. The experimental results illustrate that, with proper adjustment, both models are able to generate synthetic tmp2m anomalies that match the ground truth distribution closely.

6.2 Generative Models

In this section, we describe the two generative models that we have designed for synthetic climate data generation. A climate variable on a given date and spatial region can be seen as a two-dimensional (2D) image. The generative models target generating synthetic images of a climate variable, which can be connected to image modeling in computer vision.

6.2.1 ViT-based VAE

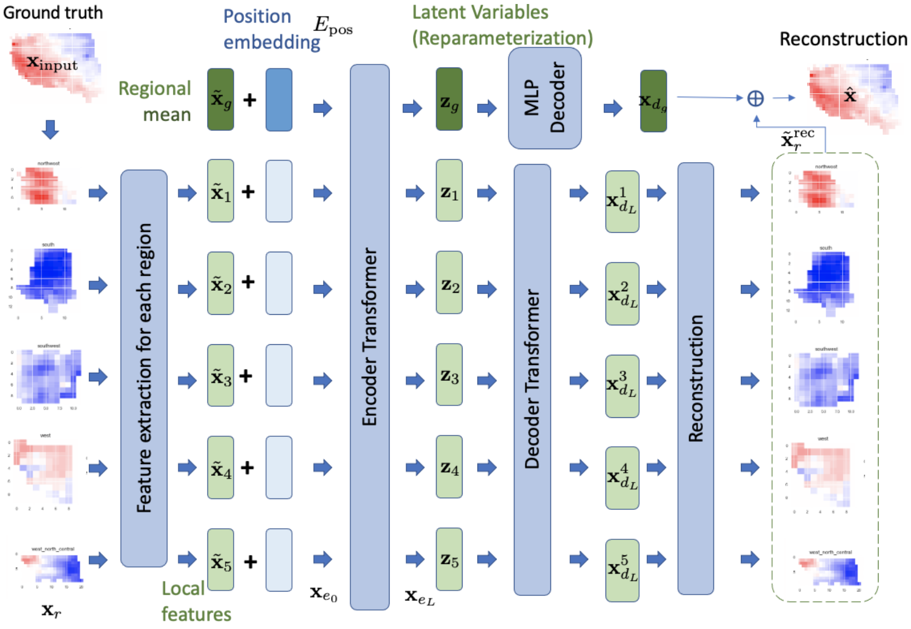


Figure 6.1: Architectures of the proposed model. The model consists of five modules: feature extraction, Transformer-based encoder, latent variables reparameterization, decoder, and reconstruction.

Inspired by the successful applications of the Visual Transformer (Dosovitskiy et al., 2020) in computer vision, we design a ViT-based VAE model for generating synthetic climate data. Figure 6.1 depicts the overview of the proposed model. The input of the model is the 2 meter temperature anomalies over the western U.S., which can be seen as a 2D image with the resolution as (H, W). More specifically, we reshape the image $\mathbf{x} \in \mathbb{R}^{H \times W}$ into a sequence of 2D patches $[\mathbf{x}_1; \cdots; \mathbf{x}_r; \cdots, ; \mathbf{x}_R]$ based on the climatically consistent regions identified by climate scientists from National Centers for Environmental Information (Figure 5.4). The output of the model is the reconstruction of $\mathbf{x}$, which is denoted as $\hat{\mathbf{x}}$. The model consists of five modules, that is feature extraction, Transformer-based encoder, (latent variables) reparameterization, decoder, and reconstruction.

Feature extraction. We consider features from two levels of spatial resolutions, i.e, local features and global features. Local features are extracted from the (high-resolution) temperature anomalies within each region, while global features represent the mean temperature anomalies of each region. For local features, we flatten the temperature anomalies over each region (denoted as $\mathbf{x}_r \in \mathbb{R}^{D_r}$) and map the vector to D_e dimensions with a trainable projection. For the r-th region, the output of the projection is referred to as the patch embedding,

$$\tilde{x}_r = \text{ReLU}(\mathbf{x}_r E_{l_1}^r) E_{l_2}^r + E_{\text{pos}}^r \ , \tag{6.1}$$

where $E_{l_1}^r \in \mathbb{R}^{D_r \times D_h}$ and $E_{l_2}^r \in \mathbb{R}^{D_h \times D_e}$ are the weight matrices, $E_{\text{pos}}^r \in \mathbb{R}^{D_e}$ is the position embedding, and Rectified Linear Unit (ReLU) is used as activation function. We add each position embedding E_{pos}^r to its corresponding patch embedding for maintaining positional information. The global feature $\tilde{\mathbf{x}}_g$ is a trainable linear projection of the regional mean $\mathbf{x}_g \in \mathbb{R}^R$,

$$\tilde{\mathbf{x}}_g = \mathbf{x}_g E_g + E_{\text{pos}}^g \ , \tag{6.2}$$

where $E_g \in \mathbb{R}^{R \times D_e}$ is the weight matrix, and E_{pos}^g is the position embedding for the global features. To create the input sequence $\mathbf{x}_{e_0}$ for the Transformer-based encoder, we concatenate global features and local features and pass them through a dropout layer, that is

$$\mathbf{x}_{e_0} = \text{Dropout}([\tilde{\mathbf{x}}_g; \tilde{\mathbf{x}}_1; \cdots; \tilde{\mathbf{x}}_r; \cdots; \tilde{\mathbf{x}}_R]) \ , \tag{6.3}$$

where $R = 5$ represents the number of regions in the western U.S. and $R + 1$ serves as the effective input sequence length for the Transformer-based encoder.

Transformer-based encoder. The encoder (Vaswani et al., 2017) is composed of multiheaded self-attention (MHSA) and multilayer perceptron (MLP) blocks. Layernorm (LN) is applied for the input of each block and there is residual connection for the output of each block. More specifically, the module is computed as

$$\mathbf{x}'_{e_l} = \text{MHSA}(\text{LN}(\mathbf{x}_{e_{l-1}})) + \mathbf{x}_{e_{l-1}} \quad \text{and} \quad \mathbf{x}_{e_l} = \text{MLP}(\text{LN}(\mathbf{x}'_{e_l})) + \mathbf{x}'_{e_l} \ , \tag{6.4}$$

where L is the number of layers ($l = 1, \cdots, L$). The MLP contains two fully-connected layers with Gaussian Error Linear Unit (GELU) as activation function.

Reparameterization. There is one latent variable $\mathbf{z}_r$ for each region (seen as the local features) and one latent variable $\mathbf{z}_g$ for the global features. The module estimates the mean and variance of each latent variable as

$$\mu_r = \mathbf{x}_{e_L}^r W_\mu^r \quad \text{and} \quad \log \text{var}(\mathbf{z}_r) = \mathbf{x}_{e_L}^r W_{\text{var}}^r \ , \tag{6.5}$$

where $W_\mu, W_{\text{var}} \in \mathbb{R}^{D_e \times D_z}$ and $r = 0, \cdots, R$. We reparameterize the latent variables using a differentiable transformation,

$$\mathbf{z}_r = \mu_r + \epsilon \odot \sigma_r = \mu_r + \epsilon \odot \exp(0.5 * \log \text{var}(\mathbf{z}_r)) \ , \tag{6.6}$$

where $\epsilon \sim \mathcal{N}(0, I)$. The global latent variable $\mathbf{z}_g = \mathbf{z}_0$ has the index as $r = 0$.

Decoder. The decoder has two components, one for local latent variables $\mathbf{x}_{d_0} = [\mathbf{z}_1; \cdots; \mathbf{z}_R]$, and one for global latent variable $\mathbf{z}_g$. The local decoder is a Transformer model with the same layer as the Transformer-based encoder. The input sequence length is R and dimension is D_z. The output of the local decoder is $\mathbf{x}_{d_L} = [\mathbf{x}_{d_L}^1; \cdots; \mathbf{x}_{d_L}^R]$. The global decoder is a MLP with batch normalization, that is

$$\mathbf{x}_{d_g} = \text{BatchNorm}(\mathbf{z}_g W_{d_1}) W_{d_2} \ , \tag{6.7}$$

with $W_{d_1} \in \mathbb{R}^{D_z \times D_h}$ and $W_{d_2} \in \mathbb{R}^{D_h \times R}$.

Reconstruction. The output of the decoder is used for reconstructing $\mathbf{x}$. For each region, the reconstruction is computed as the summation of local reconstruction and global reconstruction,

$$\hat{\mathbf{x}}_r = \tilde{\mathbf{x}}_r^{\text{rec}} + \tilde{\mathbf{x}}_{g_r} \odot I_{D_r} \ , \tag{6.8}$$

where $\tilde{\mathbf{x}}_{g_r}$ is the r-th element in $\mathbf{x}_{d_g}$ representing the regional mean for r-th region, and $I_{D_r} \in \mathbb{R}^{D_r}$ is a vector with only 1 as elements. The local reconstruction $\tilde{\mathbf{x}}_r^{\text{rec}}$ is computed as

$$\tilde{\mathbf{x}}_r^{\text{rec}} = \text{ReLU}(\mathbf{x}_{d_L}^r W_{\text{rec}_1}) W_{\text{rec}_2} \tag{6.9}$$

with $W_{\text{rec}_1} \in \mathbb{R}^{D_z \times D_h}$ and $W_{\text{rec}_2} \in \mathbb{R}^{D_h \times D_r}$. The reconstruction $\hat{\mathbf{x}} \in \mathbb{R}^{H \times W}$ is constructed by mapping $\hat{\mathbf{x}}_r$ from each region to the corresponding spatial locations.

Loss function. The loss function for one data point consists two parts: the reconstruction loss ℓ_{rec} and the Kullback–Leibler divergence (KL) loss ℓ_{KL},

$$\ell = \ell_{\text{rec}} + \beta * \ell_{\text{KL}} \ , \tag{6.10}$$

where β is a hyperparameter. (During the model training, β is annealing from 0 to 1.) The reconstruction loss is the summation of MSE over all regions, which is computed as

$$\ell_{\text{rec}} = \sum_{r=1}^{R} \sum_{i=1}^{D_r} (\mathbf{x}_r^{(i)} - \hat{\mathbf{x}}_r^{(i)})^2 \ , \tag{6.11}$$

where $\mathbf{x}_r^{(i)}$ and $\hat{\mathbf{x}}_r^{(i)}$ are the i-th element in $\mathbf{x}_r$ and $\hat{\mathbf{x}}_r$ respectively. We assume all the latent variables, including $\mathbf{z}_r$ $(r = 1, \cdots, R)$ and $\mathbf{z}_g$, are drawn from a prior $\mathcal{N}(0, I_{D_z})$. Therefore the KL loss is computed as

$$\ell_{\text{KL}} = \frac{1}{R+1} \sum_{r=0}^{R} \ell_{\text{KL}}^{r} = \frac{1}{R+1} \sum_{r=0}^{R} -0.5 * \sum_{i=1}^{D_z} (1 + \log((\sigma_r^{(i)})^2) - (\mu_r^{(i)})^2 - (\sigma_r^{(i)})^2) \ , \tag{6.12}$$

where $\mu_r^{(i)}$ and $\sigma_r^{(i)}$ are the i-th element in μ_r and σ_r.

6.2.2 Gradient Penalty (WGAN-GP)

There are two networks in GANs, a generator and a discriminator. The generator G takes a noise variable as input and projects it to synthetic samples, while the discriminator D is a classifier to distinguish between the generated synthetic samples and the samples from the input dataset. The generator G is optimized to produce "realistic" samples that are capable of confusing the discriminator D. Therefore, the objective function of GANs is formulated as

$$\min_G \max_D \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_r}[\log D(\mathbf{x})] + \mathbb{E}_{\tilde{\mathbf{x}} \sim \mathbb{P}_g}[\log(1 - D(\tilde{\mathbf{x}}))] \ , \tag{6.13}$$

where $\mathbb{P}_r$ is the data distribution over real data $\mathbf{x}$ and $\mathbb{P}_g$ is the distribution over generated samples $\tilde{\mathbf{x}} = G(\mathbf{z})$. $\mathbf{z}$ is the input noise variable $(\mathbf{z} \sim \mathbb{P}_z)$. If the discriminator is optimal, the objective function is equivalent to minimize the Jensen-Shannon (JS) divergence between $\mathbb{P}_g$ and $\mathbb{P}_r$ (Weng, 2019). However, the JS divergence are not continuous (differentiable) w.r.t the parameters in G. Thus, to stabilize the GAN training, Arjovsky et al. (2017) propose to use Wasserstein-1 distance instead of JS divergence in the objective function. Based on the Kantorovich-Rubinstein duality (Villani, 2009), the objective function is formulated as

$$\min_G \max_{\|f_D\|_L \leq 1} \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_r}[f_D(\mathbf{x})] - \mathbb{E}_{\tilde{x} \sim \mathbb{P}_g}[f_D(\tilde{\mathbf{x}})] \ , \tag{6.14}$$

where f_D (the "discriminator" in WGAN) is in the set of 1-Lipschitz functions. To enforce the 1-Lipschitz continuity of f_D, after each gradient update, the weights in f_D are clipped

Table 6.1: Loss of all three models on the test set.

	KL loss	Reconstruction loss	Loss
CNN-VAE	79.36	234.06	313.43
ViT-VAE (CNN)	67.63	150.18	217.80
ViT-VAE	75.12	129.22	204.34

to a small window $[-c, c]$, which results in a compact weight space. Even though clipping is an easy and practical trick for preserving a Lipschitz constraint, the training of WGAN is unstable. WGAN still suffers from generating poor samples and struggles to converge. Gulrajani et al. (2017) propose gradient penalty (WGAN-GP) to substitute weight clipping, which penalizes the norm of the gradient of the "discriminator" f_D with respect to its input.

In this work, the generator consists of multiple fully-connected linear layers. Each hidden layer is followed by batch normalization and a leaky ReLU as activation function. The generator takes a standard normal random vector as input and generated samples as output. The discriminator contains three fully-connected linear layers with leaky ReLU as activation function.

6.3 Experimental Setup and Results

6.3.1 Experimental Setup

We focus on average 2-meter temperature (tmp2m) anomalies over the western U.S. with the spatial resolution as 1 ° latitude by 1 ° longitude. The raw data is extracted from NOAA's Climate Prediction Center (CPC) Global Gridded Temperature dataset (Fan and Van den Dool, 2008). We first compute daily tmp2m anomalies at each grid point by

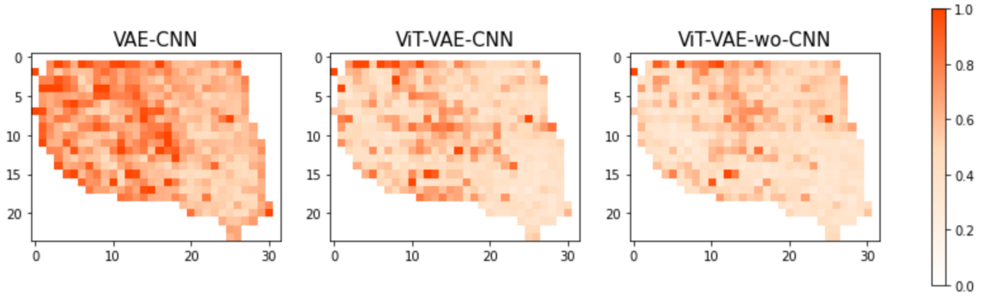
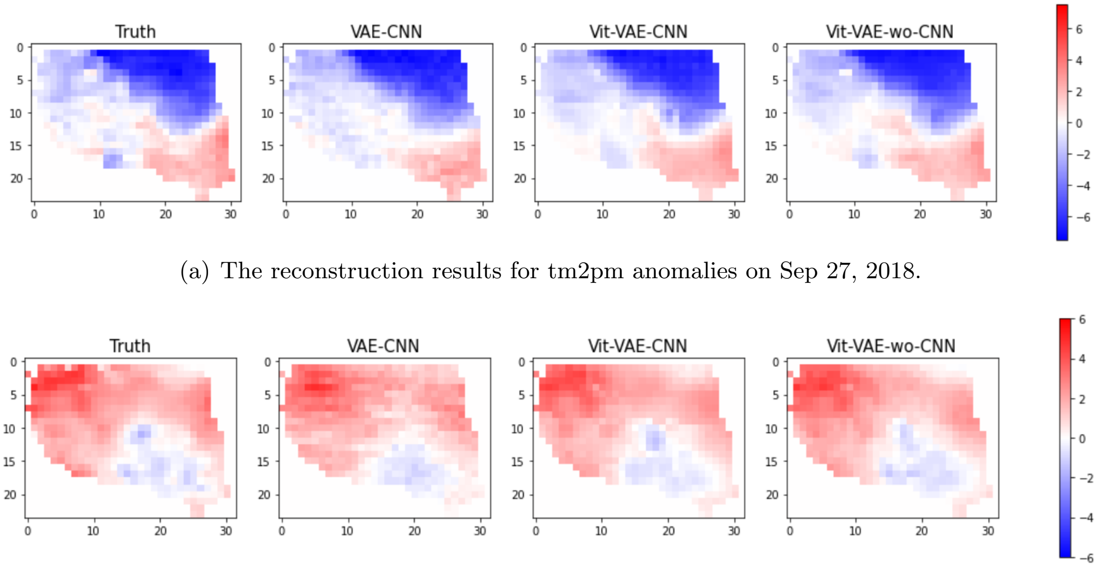


Figure 6.2: RMSE at each grid point using VAE-CNN, ViT-VAE (CNN), and the proposed ViT-VAE (without CNN). The proposed model on average achieves the best reconstruction performance.

subtracting the climatology from the observed daily tmp2m, where the climatology is the long-term average over 1986 - 2015, one for each month-day combination and grid point. The (weeks 3-4) average tmp2m anomalies computed for each date and grid point is the average of daily tmp2m anomalies for the immediate 14 days (2 weeks). The training set is constructed by the average tmp2m anomalies from 1986 to 2010 (∼ 9000 samples), while the validation set and test set consist of the data from 2011 to 2015 and the data from 2016 to 2020 respectively. The hyper-parameters are determined by the best performance over the validation set.

6.3.2 Results of ViT-based VAE



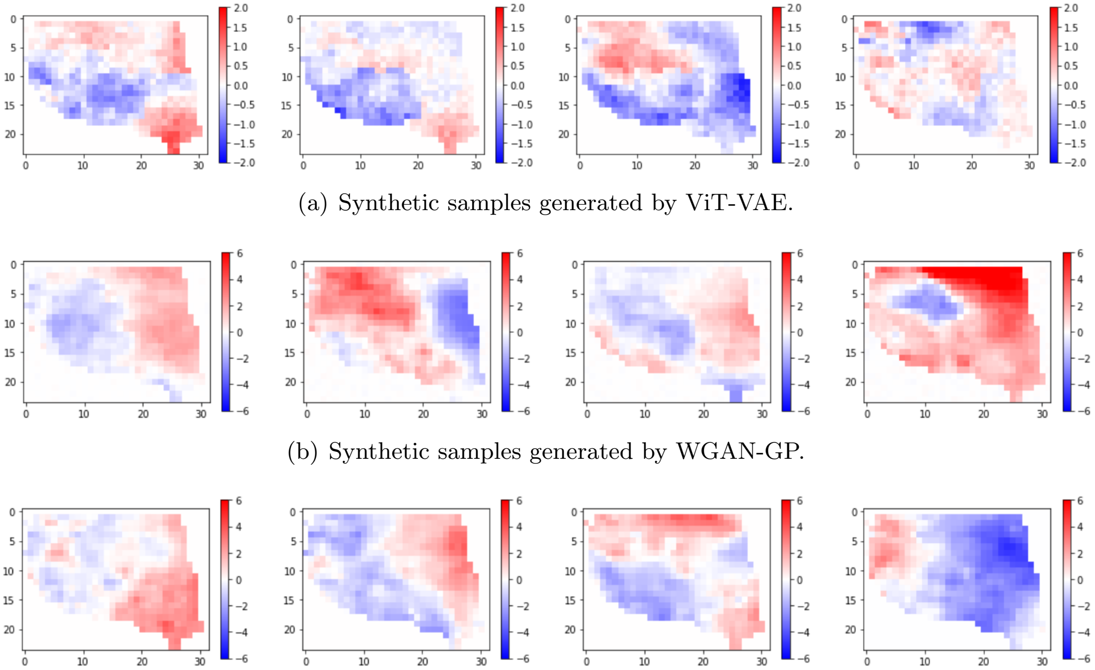
(a) The reconstruction results for tm2pm anomalies on Sep 27, 2018.

(b) The reconstruction results for tm2pm anomalies on April 4, 2016.

Figure 6.3: Two examples of tmp2m anomalies (°C) reconstruction over the western U.S. from the test set.

In this section, we show the performance of reconstruction using the ViT-based VAE (ViT-VAE) model and compare it with two baselines, VAE-CNN and ViT-VAE (CNN). VAE-CNN is a vanilla VAE model with CNN layers as its encoder and transposed CNN layers as its decoder. ViT-VAE (CNN) is a variant of the ViT-VAE model, which uses CNN layers as feature extraction and transposed CNN layers for reconstruction. Both VAE-CNN and ViT-VAE (CNN) treat the whole tmp2m anomalies image as input and do not explicitly distinguish between different regions. The loss over the test set of each model and its corresponding decomposition are listed in Table 6.1. The Vit-VAE model achieves the lowest loss on the test set and the lowest reconstruction loss as well. We also illustrate the test RMSE at each grid point in Figure 6.2. Consistent with the results on the test loss, the proposed Vit-VAE model has the best reconstruction performance with the smallest RMSE for most of the grid points compared to the two baselines. In addition, we show two anecdotal examples of reconstruction from the test set. As shown in Figure 6.3, ViT-VAE can reconstruct the spatial pattern of the ground truth tmp2m anomalies, as well as preserve some minor details and spatial smoothness, whereas VAE-CNN only roughly reconstructed the spatial pattern and lack of spatial smoothness. The experimental results indicate that significant performance gains are obtained from using the region-level feature extraction and Transformer for encoding and decoding. Nevertheless, the CNN layers are not as useful in improving the reconstruction performance as expected.

6.3.3 Synthetic Samples Generation



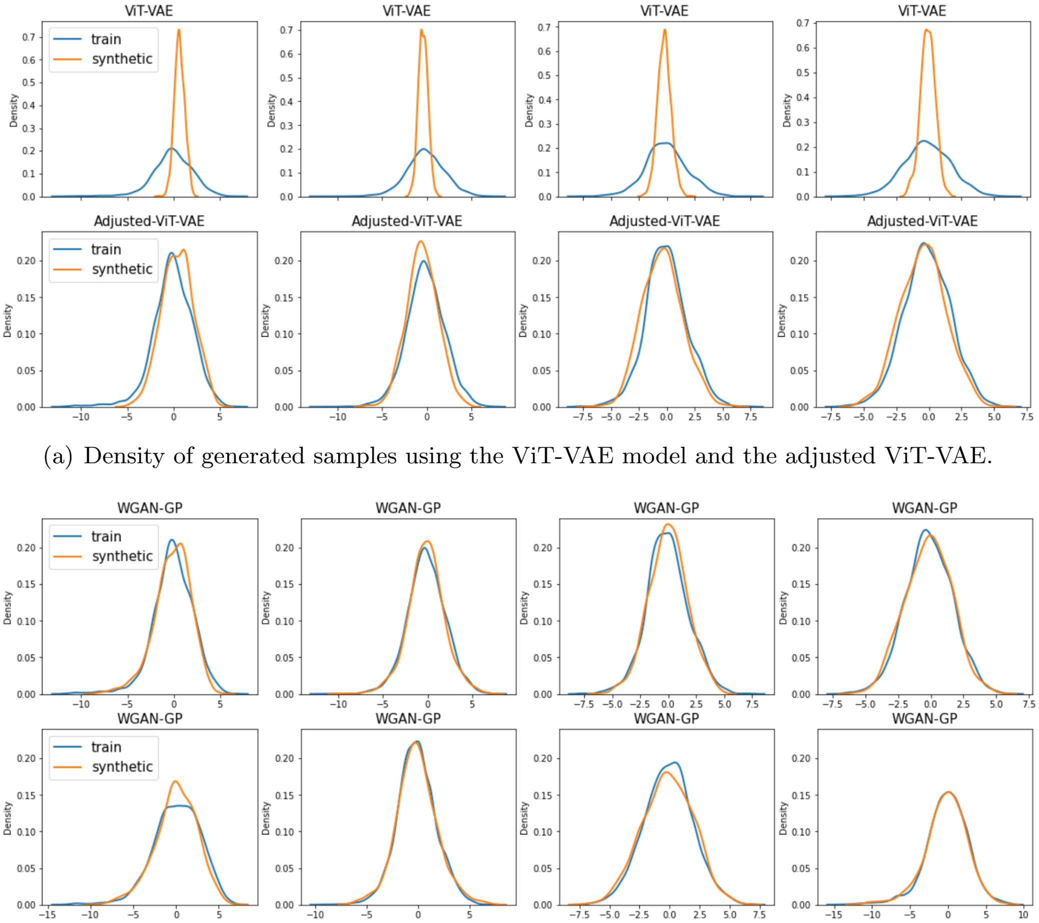
(a) Synthetic samples generated by ViT-VAE.

(b) Synthetic samples generated by WGAN-GP.

(c) Groundtruth tmp2m anomalies.

Figure 6.4: Synthetic samples generated by ViT-VAE and WGAN-GP, and the groundtruth tmp2m anomalies from the training set.

Both ViT-VAE and WGAN-GP can be used for generating synthetic samples. For ViT-VAE, we sample the latent variables from a multivariate normal distribution with independent zero mean and unit-variance components and feed them to the decoder for generating synthetic samples. Similarly, for WGAN-GP, we use unit-variance Gaussian noise as input, and the generator can produce synthetic samples. 1000 samples are generated using the two models separately. We show four randomly-selected generated samples from each model in Figure 6.4 to analyze and discuss the performance of each model. For comparison, we also present four examples of ground truth tmp2m anomalies. Visually, it is hard to distinguish the generated synthetic samples from the ground truth tmp2m anomalies, which indicates that both deep learning models are able to simulate realistic spatial patterns in tmp2m anomalies.



(a) Density of generated samples using the ViT-VAE model and the adjusted ViT-VAE.

(b) Density of generated samples using the WGAN-GP model.

Figure 6.5: Probability density function comparison between the grountruth data and the generated samples. Figure (a) and the top row in Figure (b) present the results on the same four grid points. In addition, the bottom row in Figure (b) presents the results for another four randomly-selected points.

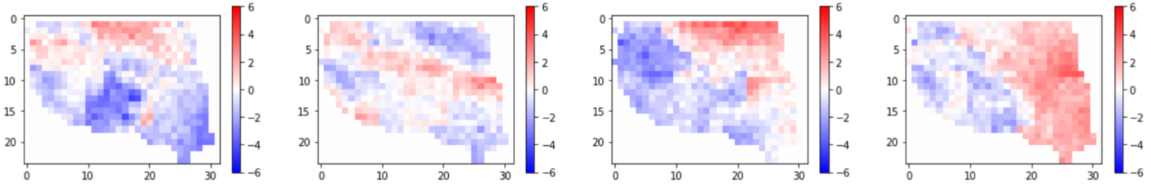


Figure 6.6: Generated synthetic samples using the adjusted ViT-VAE model.

However, the synthetic samples generated by ViT-VAE have much smaller values, mainly from −2°C to 2°C, compared to the synthetic samples from WGAN-GP and the ground truth samples that contain values varying from −6°C to 6°C. In Figure 6.5(a), we compare the probability density function of all generated samples using ViT-VAE and the training samples at four grid points. The results also verify that the variance of generated samples using ViT-VAE model is much smaller than the ground truth tmp2m anomalies. Nevertheless, as shown in Figure 6.5(b), WGAN-GP is able to generate synthetic samples that match the variance and tail distributions of the ground truth for most of the grid points. We suspect that the sampled latent variables in ViT-VAE are close to zero, which leads to the small variance of generated samples. Therefore, to fix the variance issue for ViT-VAE, instead of sampling the latent variables from a unit-variance normal distribution, we increase the variance of prior distribution to a larger constant and name it adjusted ViT-VAE. The probabilistic density functions of the newly generated samples using the adjusted ViT-VAE are shown in Figure 6.5(a) (the bottom row). By adopting the change, we have observed a significant improvement, that the density function of the newly generated samples (using the adjusted ViT-VAE) can match the ground truth distribution more closely. Four examples of the new synthetic samples are presented in Figure 6.6 for comparison. The newly generated samples match the general magnitude of ground truth tmp2m anomalies.

Since the decoder in the ViT-VAE model produces the synthetic samples on the region level, there exist some non-smooth changes over the boundaries between two adjacent regions. For future work, some improvements on preserving spatial smoothness are needed. Besides, the distribution of tmp2m anomalies varies over seasons. For example, the variances of tmp2m anomalies in winter are usually higher than the variances in summer for the central U.S. Therefore, it is worth exploring the idea of generating synthetic samples for each season separately.

6.4 Discussions

In this chapter, we focus on generating synthetic 2-meter temperature anomalies using deep generative models. We propose a novel ViT-based variational autoencoder (ViT-VAE) model which combines the state-of-the-art computer vision model with variational autoencoder. The proposed model can learn latent representations of tmp2m anomalies for each climatically consistent region and generate synthetic climate data. The experimental results illustrate that the proposed model can outperform baseline models and generate more accurate reconstruction on the test set. In addition, we carefully compare the proposed model with another popular type of generative model, i.e., WGAN-GP. With proper adjustment, both models are able to generate synthetic tmp2m anomalies that match the ground truth distribution closely. Further, we discuss the potential directions to improve the proposed ViT-VAE model for generating more realistic synthetic samples.

Part IV
Concluding Remarks

Chapter 7

Conclusions

In this dissertation, we aim on developing machine learning models for tackling two fundamental problems in climate science, which are (a) understanding the dependencies among or within key components in the Earth's climate system and (b) forecasting climate variables on sub-seasonal time scales. We propose novel machine learning models for each problem and perform rigorous empirical evaluations on synthetic and real-world climate data.

We begin with Part II, which covers two particular applications in understanding the dependencies among or within ocean, land, and atmosphere. In Chapter 2, we consider the use of structure learning methods for probabilistic graphical models to identify statistical dependencies in high-dimensional physical processes. We propose ACLIME-ADMM, an efficient two-step algorithm for adaptive structure learning, which decides a suitable edge-specific threshold in a data-driven statistically rigorous manner. We compare the proposed ACLIME-ADMM with baseline structure learning approaches on both synthetic data simulated by PDEs, and real data of daily global geopotential heights. ACLIME-ADMM is shown to be efficient, stable, and competitive, especially can outperform the baselines in challenging scenarios from synthetic data. On the real-world dataset, ACLIME-ADMM is able to recover the underlying structure of global atmospheric circulation, including switches in wind directions at the equator and tropics entirely from the data.

In Chapter 3, we focus on identifying predictive relationships between land and ocean climate variables. More specifically, we consider the problem of predicting monthly deseasonalized land temperature at different locations worldwide from sea surface temperatures. We introduce a weighted Lasso model for the problem which yields interpretable results while being highly accurate. In addition, we establish finite sample estimation error bounds for weighted Lasso, and illustrate its superior empirical performance and interpretability over some complex models, such as Deep nets and GBT. We also present a detailed empirical analysis of what has been wrong with Deep Nets for the problem, which may serve as a helpful guideline for applying Deep nets to other small sample problems in climate science.

In Part III, we shift focus towards a specific forecasting problem, sub-seasonal climate forecasting. In Chapter 4, we carefully investigate 10 machine learning approaches to sub-seasonal temperature forecasting over the contiguous U.S. Our results indicate that suitable machine learning models, e.g., XGBoost, to some extent, capture the predictability on sub-seasonal time scales and can outperform the climatological baselines, while deep learning models barely manage to match the best results with carefully designed architectures. Besides, our analysis and exploration provide insights on important aspects to improve the quality of sub-seasonal forecasts, e.g., feature representation and training set construction.

In Chapter 5, we extend our attention to existing climate models for SSF. We perform a fine-grained comparison of a suite of modern machine learning models with state-of-the-art physics-based dynamical models from the SubX project for SSF over the western U.S. Empirical results indicate that, on average, machine learning models outperform dynamical models while the machine learning models tend to generate forecasts with conservative magnitude compared to the SubX models. Further, we explore mechanisms to enhance the machine learning models and show that suitably incorporating dynamical model forecasts as inputs to machine learning models can substantially improve the forecasting performance of the machine learning models.

In Chapter 6, to compensate for the limited availability of high-resolution climate data for SSF, we seek to generate synthetic tmp2m anomalies using deep generative models. We propose a novel ViT-based VAE model which uses the state-of-the-art computer vision model as the encoder and decoder of VAE. The proposed model can learn latent representations of tmp2m anomalies for each climatically consistent region, and generate synthetic climate data over the western U.S. In addition, we carefully compare the proposed model with another popular type of generative model, i.e., WGAN-GP. Empirical results illustrate that both models are capable of generating synthetic tmp2m anomalies that match the ground truth distribution closely.

In summary, this dissertation presents empirical and theoretical work on tackling critical problems in climate science using machine learning techniques. We hope that the presented work will lead to continued progress in integrating machine learning into climate science for improving the interpretability in climate modeling and advancing the quality of sub-seasonal climate forecasts.

References

NOAA - National Centers for Environmental Information (2021). U.S. Climate Regions. `https://www.ncdc.noaa.gov/monitoring-references/maps/us-climate-regions.php`.

Aalto, J., Pirinen, P., Heikkinen, J., and Venäläinen, A. (2013). Spatial interpolation of monthly climate data for finland: comparing the performance of kriging and generalized additive models. *Theoretical and Applied Climatology*, 112(1-2):99–111.

Alemi, A., Poole, B., Fischer, I., Dillon, J., Saurous, R. A., and Murphy, K. (2018). Fixing a broken elbo. In *International Conference on Machine Learning*, pages 159–168.

An, J. and Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1):1–18.

Arcomano, T., Szunyogh, I., Pathak, J., Wikner, A., Hunt, B. R., and Ott, E. (2020). A machine learning-based global atmospheric forecast model. *Geophysical Research Letters*, 47(9).

Arjovsky, M. and Bottou, L. (2017). Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR.

Badr, H. S., Zaitchik, B. F., and Guikema, S. D. (2014). Application of statistical models to the prediction of seasonal rainfall anomalies over the sahel. *Journal of Applied meteorology and climatology*, 53(3):614–636.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

Banerjee, O., Ghaoui, L. E., and d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516.

Barnston, A. G., Tippett, M. K., L'Heureux, M. L., Li, S., and DeWitt, D. G. (2012). Skill of real-time seasonal enso model predictions during 2002–11: Is our capability increasing? *Bulletin of the American Meteorological Society*, 93(5):631–651.

Bendito, E., Carmona, A., Encinas, A. M., and Gesto, J. M. (2007). Estimation of fekete points. *Journal of Computational Physics*, 225(2):2354–2376.

Bengio, Y. (2012). *Practical Recommendations for Gradient-Based Training of Deep Architectures*, pages 437–478. Springer Berlin Heidelberg, Berlin, Heidelberg.

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

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. (2015). Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. (2016). Generating sentences from a continuous space. In *20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016*, pages 10–21. Association for Computational Linguistics (ACL).

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1).

Braman, L. M., van Aalst, M. K., Mason, S. J., Suarez, P., Ait-Chellouche, Y., and Tall, A. (2013). Climate forecasts in disaster management: Red cross flood operations in west africa, 2008. *Disasters*, 37(1):144–164.

Buchmann, P. and DelSole, T. (2021). Week 3-4 prediction of wintertime conus temperature using machine learning techniques. *Frontiers in Climate*, 3:81.

Burda, Y., Grosse, R. B., and Salakhutdinov, R. (2016). Importance weighted autoencoders. In *International Conference on Learning Representations.*

Butler, A. H., Seidel, D. J., Hardiman, S. C., Butchart, N., Birner, T., and Match, A. (2015). Defining sudden stratospheric warmings. *Bulletin of the American Meteorological Society*, 96(11):1913 – 1928.

Cai, L., Gao, H., and Ji, S. (2019). Multi-stage variational auto-encoders for coarse-to-fine image generation. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 630–638. SIAM.

Cai, T., Liu, W., and Luo, X. (2011). A constrained L1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607.

Cai, T. T., Liu, W., Zhou, H. H., et al. (2016). Estimating sparse precision matrix: Optimal rates of convergence and adaptive estimation. *The Annals of Statistics*, 44(2):455–488.

Chandrasekaran, V., Recht, B., Parrilo, P. A., and Willsky, A. S. (2012). The convex geometry of linear inverse problems. *Foundations of Computational mathematics,*, 12(6):805–849.

Chatterjee, S., Steinhaeuser, K., Banerjee, A., Chatterjee, S., and Ganguly, A. (2012). Sparse group lasso: Consistency and climate applications. In *Proceedings of the 2012 SIAM International Conference on Data Mining (SDM)*, pages 47–58.

Chen, R. T., Li, X., Grosse, R. B., and Duvenaud, D. K. (2018). Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31.

Chen, S. and Banerjee, A. (2015). Structured estimation with atomic norms: General bounds and applications. In *Advances in Neural Information Processing Systems*, pages 2908–2916.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 785–794.

Chen, X.-W., Anantha, G., and Wang, X. (2006). An effective structure learning method for constructing gene networks. *Bioinformatics*, 22(11):1367–1374.

Cheng, J., Liu, J., Kuang, Q., Xu, Z., Shen, C., Liu, W., and Zhou, K. (2021). Deepdt: Generative adversarial network for high-resolution climate prediction. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5.

Child, R. (2020). Very deep vaes generalize autoregressive models and can outperform them on images. In *International Conference on Learning Representations*.

Chollet, F. (2015). Keras. `https://github.com/fchollet/keras`.

Chu, T., Danks, D., and Glymour, C. (2005). Data driven methods for nonlinear granger causality: Climate teleconnection mechanisms. Technical report, Carnegie Mellon University.

Cofino, A. S., Cano, R., Sordo, C., and Gutiérrez, J. M. (2002). Bayesian networks for probabilistic weather prediction. In *In Proceedings of the 15th Eureopean Conference on Artificial Intelligence (ECAI)*, pages 695–699.

Cohen, J., Coumou, D., Hwang, J., Mackey, L., Orenstein, P., Totz, S., and Tziperman, E. (2019). S2s reboot: An argument for greater inclusion of machine learning in subseasonal to seasonal forecasts. *Wiley Interdisciplinary Reviews: Climate Change*, 10(2):e00567.

Colombo, D. and Maathuis, M. H. (2014). Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 15(1):3741–3782.

Cremer, C., Li, X., and Duvenaud, D. (2018). Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning*, pages 1078–1086. PMLR.

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., and Bharath, A. A. (2018). Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65.

Daniel, W. W. (1978). *Applied nonparametric statistics*. Houghton Mifflin.

de Perez, E. C. and Mason, S. J. (2014). Climate information for humanitarian agencies: Some basic principles. *Earth Perspectives*, 1(1):11.

DelSole, T. and Banerjee, A. (2017). Statistical seasonal prediction based on regularized regression. *Journal of Climate,*, 30(4):1345–1361.

Delsole, T. and Tippett, M. (2017). Predictability in a changing climate. *Climate Dynamics*.

DelSole, T. and Tippett, M. K. (2016). Forecast comparison based on random walks. *Monthly Weather Review*, 144(2):615–626.

DelSole, T., Trenary, L., Tippett, M. K., and Pegion, K. (2017). Predictability of week-3–4 average temperature and precipitation over the contiguous united states. *Journal of Climate*, 30(10):3499 – 3512.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Li, F.-F. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Deng, Y. and Ebert-Uphoff, I. (2014). Weakening of atmospheric information flow in a warming climate in the community climate system model. *Geophysical Research Letters*, 41(1):193–200.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Domeisen, D. I. and Butler, A. H. (2020). Stratospheric drivers of extreme events at the earth's surface. *Communications Earth & Environment*, 1(1):1–8.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Drton, M. and Maathuis, M. H. (2017). Structure learning in graphical modeling. *Annual Review of Statistics and Its Application*, 4:365–393.

Dueben, P. D. and Bauer, P. (2018). Challenges and design choices for global weather and climate models based on machine learning. *Geoscientific Model Development*, 11(10):3999–4009.

Ebert-Uphoff, I. and Deng, Y. (2012). A new type of climate network based on probabilistic graphical models: Results of boreal winter versus summer. *Geophysical Research Letters*, 39(19).

Ebert-Uphoff, I. and Deng, Y. (2014). Causal discovery from spatio-temporal data with applications to climate science. In *International Conference on Machine Learning and Applications*, pages 606–613. IEEE.

Edwards, P. N. (2011). History of climate modeling. *Wiley Interdisciplinary Reviews: Climate Change*, 2(1):128–139.

Fan, Y. and van den Dool, H. (2004). Climate prediction center global monthly soil moisture data set at 0.5 resolution for 1948 to present. *Journal of Geophysical Research: Atmospheres*, 109(D10).

Fan, Y. and Van den Dool, H. (2008). A global monthly land surface air temperature analysis for 1948–present. *Journal of Geophysical Research: Atmospheres*, 113(D1).

Fanfarillo, A., Roozitalab, B., Hu, W., and Cervone, G. (2021). Probabilistic forecasting using deep generative models. *GeoInformatica*, 25(1):127–147.

Flato, G. M. (2011). Earth system models: an overview. *Wiley Interdisciplinary Reviews: Climate Change*, 2(6):783–800.

Fogli, P. G., Manzini, E., Vichi, M., Alessandri, A., Patara, L., Gualdi, S., Scoccimarro, E., Masina, S., and Navarra, A. (2009). Ingv-cmcc carbon (icc): A carbon cycle earth system model. *CMCC Research Paper,*, 61:31.

Francis, R. and Renwick, J. (1998). A regression-based assessment of the predictability of new zealand climate anomalies. *Theoretical and Applied Climatology,*, 60(1):21–36.

Frederik Nebeker (1995). *Calculating the weather: Meteorology in the 20th century.* Elsevier.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Funahashi, K.-i. and Nakamura, Y. (1993). Approximation of dynamical systems by continuous time recurrent neural networks. *Neural networks*, 6(6):801–806.

Gelaro, R., McCarty, W., Suárez, M. J., Todling, R., Molod, A., Takacs, L., Randles, C. A., Darmenov, A., Bosilovich, M. G., Reichle, R., et al. (2017). The modern-era retrospective analysis for research and applications, version 2 (merra-2). *Journal of climate*, 30(14):5419–5454.

Gers, F. A., Schmidhuber, J., and Cummins, F. (2000). Learning to forget: Continual prediction with lstm. *Neural Computation*, 12(10):2451–2471.

Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452–459.

Golmohammadi, J., Ebert-Uphoff, I., He, S., Deng, Y., and Banerjee, A. (2017). High-dimensional dependency structure learning for physical processes. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 883–888.

Goncalves, A. R., Banerjee, A., and Von Zuben, F. J. (2017). Spatial projection of multiple climate variables using hierarchical multitask learning. In *AAAI Conference on Artificial Intelligence*.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning.* MIT press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Goovaerts, P. (1999). Geostatistics in soil science: state-of-the-art and perspectives. *Geoderma*, 89(1-2):1–45.

Gordon, Y. (1988). On milman's inequality and random subspaces which escape through a mesh in r^n. In *Geometric Aspects of Functional Analysis*, pages 84–106. Springer.

Gregor, K., Danihelka, I., Graves, A., Rezende, D., and Wierstra, D. (2015). Draw: A recurrent neural network for image generation. In *International Conference on Machine Learning*, pages 1462–1471. PMLR.

Grover, A., Kapoor, A., and Horvitz, E. (2015). A deep hybrid model for weather forecasting. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 379–386. ACM.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved training of wasserstein gans. *Advances in neural information processing systems*, 30.

Ham, Y.-G., Kim, J.-H., and Luo, J.-J. (2019). Deep learning for multi-year enso forecasts. *Nature*, 573(7775):568–572.

Harris, N. and Drton, M. (2013). Pc algorithm for nonparanormal graphical models. *Journal of Machine Learning Research*, 14(1):3365–3383.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778.

He, S., Li, X., DelSole, T., Ravikumar, P., and Banerjee, A. (2021a). Sub-seasonal climate forecasting via machine learning: Challenges, analysis, and advances. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):169–177.

He, S., Li, X., Sivakumar, V., and Banerjee, A. (2019). Interpretable predictive modeling for climate variables with weighted lasso. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1385–1392.

He, S., Li, X., Trenary, L., Cash, B. A., DelSole, T., and Banerjee, A. (2021b). Learning and dynamical models for sub-seasonal climate forecasting: Comparison and collaboration. *arXiv preprint arXiv:2110.05196*.

He, S., Li, X., Trenary, L., Cash, B. A., DelSole, T., and Banerjee, A. (2021c). Machine learning and dynamical models for sub-seasonal climate forecasting. *NeurIPS workshop on Machine Learning and the Physical Sciences*.

Higgins, I., Matthey, L., Pal, A., Burgess, C. P., Glorot, X., Botvinick, M. M., Mohamed, S., and Lerchner, A. (2017). beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Hourdin, F., Mauritsen, T., Gettelman, A., Golaz, J.-C., Balaji, V., Duan, Q., Folini, D., Ji, D., Klocke, D., Qian, Y., et al. (2017). The art and science of climate model tuning. *Bulletin of the American Meteorological Society*, 98(3):589–602.

Hsieh, C.-J., Dhillon, I. S., Ravikumar, P. K., and Sustik, M. A. (2011). Sparse inverse covariance matrix estimation using quadratic approximation. In *Advances in Neural Information Processing Systems*, pages 2330–2338.

Hsieh, W. W. and Tang, B. (1998). Applying neural network models to prediction and data analysis in meteorology and oceanography. *Bulletin of the American Meteorological Society,*, 79(9):1855–1870.

Hu, Z., Yang, Z., Salakhutdinov, R., and Xing, E. P. (2018). On unifying deep generative models. In *International Conference on Learning Representations.*

Huang, J., Ma, S., and Zhang, C.-H. (2008). Adaptive lasso for sparse high-dimensional regression models. *Statistica Sinica,*, pages 1603–1618.

Hwang, J., Orenstein, P., Cohen, J., Pfeiffer, K., and Mackey, L. (2019). Improving subseasonal forecasting in the western us with machine learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2325–2335. ACM.

Jalali, A., Ravikumar, P., and Sanghavi, S. (2013). A dirty model for multiple sparse regression. *IEEE Transactions on Information Theory*, 59(12):7947–7968.

Jolliffe, I. (2011). Principal component analysis. In *International encyclopedia of statistical science*, pages 1094–1096. Springer.

Jordan, M. I. and Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260.

Kalisch, M. and Bühlmann, P. (2007). Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *Journal of Machine Learning Research*, 8(Mar):613–636.

Kalnay, E., Kanamitsu, M., Kistler, R., Collins, W., Deaven, D., Gandin, L., Iredell, M., Saha, S., White, G., Woollen, J., et al. (1996a). The ncep/ncar 40-year reanalysis project. *Bulletin of the American meteorological Society*, 77(3):437–471.

Kalnay, E., Kanamitsu, M., Kistler, R., Collins, W., Deaven, D., Gandin, L., Iredell, M., Saha, S., White, G., Woollen, J., et al. (1996b). The ncep/ncar 40-year reanalysis project. *Bulletin of the American meteorological Society*, 77(3):437–472.

Karl, T. and Koss, W. J. (1984). Regional and national monthly, seasonal, and annual temperature weighted by area, 1895-1983. *Historical climatology series 4-3.*

Karpatne, A., Babaie, H. A., Ravela, S., Kumar, V., and Ebert-Uphoff, I. (2017). Machine learning for the geosciences - opportunities, challenges, and implications for the ML process. In *SIAM SDM 2017, Workshop on Mining Big Data in Climate and Environment.*

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018). Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations.*

Kingma, D. P. (2017). Variational inference & deep learning: A new synthesis. *PhD Thesis.*

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

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2016). Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29.

Kingma, D. P. and Welling, M. (2014). Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*.

Kingma, D. P., Welling, M., et al. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392.

Klemm, T. and McPherson, R. A. (2017). The development of seasonal climate forecasting for agricultural producers. *Agricultural and forest meteorology*, 232:384–399.

Koster, R. D., Mahanama, S. P. P., Yamada, T. J., Balsamo, G., Berg, A. A., Boisserie, M., Dirmeyer, P. A., Doblas-Reyes, F. J., Drewitt, G., Gordon, C. T., Guo, Z., Jeong, J.-H., Lee, W.-S., Li, Z., Luo, L., Malyshev, S., Merryfield, W. J., Seneviratne, S. I., Stanelle, T., van den Hurk, B. J. J. M., Vitart, F., and Wood, E. F. (2011). The second phase of the global land–atmosphere coupling experiment: Soil moisture contributions to subseasonal forecast skill. *Journal of Hydrometeorology*, 12(5):805 – 822.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012a). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105.

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

Kuligowski, R. J. and Barros, A. P. (1998). Localized precipitation forecasts from a numerical weather prediction model using artificial neural networks. *Weather and forecasting*, 13(4):1194–1204.

Kunsch, H. R. (1989). The jackknife and the bootstrap for general stationary observations. *The annals of Statistics*, pages 1217–1241.

Kusner, M. J., Paige, B., and Hernández-Lobato, J. M. (2017). Grammar variational autoencoder. In *International Conference on Machine Learning*, pages 1945–1954. PMLR.

Lauritzen, S. L. (1996). *Graphical models*, volume 17. Clarendon Press.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature,*, 521(7553):436.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE,*, 86(11):2278–2324.

Li, S. and Robertson, A. W. (2015). Evaluation of submonthly precipitation forecast skill from global ensemble prediction systems. *Monthly Weather Review*, 143(7):2871–2889.

Lipovetsky, S. and Conklin, M. (2001). Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17(4):319–330.

Liu, H., Han, F., Yuan, M., Lafferty, J., Wasserman, L., et al. (2012). High-dimensional semiparametric gaussian copula graphical models. *The Annals of Statistics*, 40(4):2293–2326.

Liu, Y., Racah, E., Correa, J., Khosrowshahi, A., Lavers, D., Kunkel, K., Wehner, M., Collins, W., et al. (2016). Application of deep convolutional neural networks for detecting extreme weather in climate datasets. *arXiv preprint arXiv:1605.01156*.

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. (2019). Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR.

Lorenc, A. C. (1986). Analysis methods for numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 112(474):1177–1194.

Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of atmospheric sciences*, 20(2):130–141.

Lucas, J., Tucker, G., Grosse, R. B., and Norouzi, M. (2019). Don't blame the elbo! a linear vae perspective on posterior collapse. *Advances in Neural Information Processing Systems*, 32.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774.

Maaløe, L., Fraccaro, M., Liévin, V., and Winther, O. (2019). Biva: A very deep hierarchy of latent variables for generative modeling. *Advances in neural information processing systems*, 32.

Mariotti, A., Baggett, C., Barnes, E. A., Becker, E., Butler, A., Collins, D. C., Dirmeyer, P. A., Ferranti, L., Johnson, N. C., Jones, J., et al. (2020). Windows of opportunity for skillful forecasts subseasonal to seasonal and beyond. *Bulletin of the American Meteorological Society*, 101(5):E608–E625.

Masrani, V., Le, T. A., and Wood, F. (2019). The thermodynamic variational objective. *Advances in Neural Information Processing Systems*, 32.

Masters, D. and Luschi, C. (2018). Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*.

McDermott, P. L. and Wikle, C. K. (2017). Bayesian recurrent neural network models for forecasting and quantifying uncertainty in spatial-temporal data. *arXiv preprint arXiv:1711.00636*.

McGovern, A., Gagne, D. J., Williams, J. K., Brown, R. A., and Basara, J. B. (2014). Enhancing understanding and improving prediction of severe weather through spatiotemporal relational learning. *Machine learning*, 95(1):27–50.

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of statistics*, pages 1436–1462.

Meinshausen, N. and Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473.

Metz, L., Poole, B., Pfau, D., and Sohl-Dickstein, J. (2016). Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*.

Mouatadid, S., Orenstein, P., Flaspohler, G., Oprescu, M., Cohen, J., Wang, F., Knight, S., Geogdzhayeva, M., Levang, S., Fraenkel, E., et al. (2021). Learned benchmarks for subseasonal forecasting. *arXiv preprint arXiv:2109.10399*.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning*, pages 807–814.

National Academies of Sciences, Engineering, and Medicine (2016). *Next generation earth system prediction: strategies for subseasonal to seasonal forecasts*. National Academies Press.

National Research Council (2010). *Assessment of intraseasonal to interannual climate prediction and predictability*. National Academies Press.

Negahban, S. N., Ravikumar, P., Wainwright, M. J., and Yu, B. (2012). A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557.

NIDIS (2019). Forecast Rodeo II Leaderboard. `https://www.drought.gov/forecast-rodeo-ii-leaderboard`.

NOAA (2021). Billion-Dollar Weather and Climate Disasters. `https://www.ncdc.noaa.gov/billions/events`.

O'Brien, G., O'Keefe, P., Rose, J., and Wisner, B. (2006). Climate change and disaster management. *Disasters*, 30(1):64–80.

Olivieri, A. C. (2018). *Introduction to Multivariate Calibration: A Practical Approach*. Springer.

Pachauri, R. K., Allen, M. R., Barros, V. R., Broome, J., Cramer, W., Christ, R., Church, J. A., Clarke, L., Dahe, Q., Dasgupta, P., et al. (2014). *Climate change 2014: synthesis report. Contribution of Working Groups I, II and III to the fifth assessment report of the Intergovernmental Panel on Climate Change*. IPCC.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman, 2nd edition.

Pearl, J. (2009). *Causality: Models, Reasoning, and Inference*. Cambridge university press.

Pegion, K., Kirtman, B. P., Becker, E., Collins, D. C., LaJoie, E., Burgman, R., Bell, R., DelSole, T., Min, D., Zhu, Y., Li, W., Sinsky, E., Guan, H., Gottschalck, J., Metzger, E. J., Barton, N. P., Achuthavarier, D., Marshak, J., Koster, R. D., Lin, H., Gagnon, N., Bell, M., Tippett, M. K., Robertson, A. W., Sun, S., Benjamin, S. G., Green, B. W., Bleck, R., and Kim, H. (2019). The subseasonal experiment (subx): A multimodel subseasonal prediction experiment. *Bulletin of the American Meteorological Society*, 100(10):2043–2060.

Pomeroy, J., Gray, D., Hedstrom, N., and Janowicz, J. (2002). Prediction of seasonal snow accumulation in cold climate forecasts. *Hydrological Processes*, 16(18):3543–3558.

Poole, B., Alemi, A. A., Sohl-Dickstein, J., and Angelova, A. (2016). Improved generator objectives for gans. *arXiv preprint arXiv:1612.02780*.

Pu, Y., Gan, Z., Henao, R., Yuan, X., Li, C., Stevens, A., and Carin, L. (2016). Variational autoencoder for deep learning of images, labels and captions. *Advances in neural information processing systems*, 29:2352–2360.

Racah, E., Beckham, C., Maharaj, T., Ebrahimi Kahou, S., Prabhat, M., and Pal, C. (2017). Extremeweather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. *Advances in neural information processing systems*, 30.

Radhika, Y. and Shashi, M. (2009). Atmospheric temperature prediction using support vector machines. *International journal of computer theory and engineering*, 1(1):55.

Raff, D., Nowak, K., Cifelli, R., Brekke, L. D., and Webb, R. S. (2017). Sub-seasonal climate forecast rodeo. In *AGU Fall Meeting*.

Ravuri, S., Lenc, K., Willson, M., Kangin, D., Lam, R., Mirowski, P., Fitzsimons, M., Athanassiadou, M., Kashem, S., Madge, S., et al. (2021). Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677.

Razavi, A., van den Oord, A., Poole, B., and Vinyals, O. (2018). Preventing posterior collapse with delta-vaes. In *International Conference on Learning Representations*.

Reichle, R. H. and Liu, Q. (2014). Observation-corrected precipitation estimates in geos-5. *Technical Report Series on Global Modelling and Data Assimilation*, 35:24.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., et al. (2019). Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204.

Reshef, D. N., Reshef, Y. A., Finucane, H. K., Grossman, S. R., McVean, G., Turnbaugh, P. J., Lander, E. S., Mitzenmacher, M., and Sabeti, P. C. (2011). Detecting novel associations in large data sets. *science*, 334(6062):1518–1524.

Reynolds, R. W., Smith, T. M., Liu, C., Chelton, D. B., Casey, K. S., and Schlax, M. G. (2007). Daily high-resolution-blended analyses for sea surface temperature. *Journal of Climate*, 20(22):5473–5496.

Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR.

Rolfe, J. T. (2016). Discrete variational autoencoders. *arXiv preprint arXiv:1609.02200*.

Rolnick, D., Donti, P. L., Kaack, L. H., Kochanski, K., Lacoste, A., Sankaran, K., Ross, A. S., Milojevic-Dupont, N., Jaques, N., Waldman-Brown, A., et al. (2019). Tackling climate change with machine learning. *arXiv preprint arXiv:1906.05433.*

Romm, J. (2018). *Climate Change: What Everyone Needs to Know®.* Oxford University Press.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision,*, 115(3):211–252.

Saha, S., Moorthi, S., Wu, X., Wang, J., Nadiga, S., Tripp, P., Behringer, D., Hou, Y.-T., Chuang, H.-y., Iredell, M., et al. (2014). The ncep climate forecast system version 2. *Journal of climate*, 27(6):2185–2208.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans. *Advances in neural information processing systems*, 29.

Scher, S. and Messori, G. (2019). Weather and climate forecasting with neural networks: using general circulation models (gcms) with different complexity as a study ground. *Geoscientific Model Development*, 12(7):2797–2809.

Schneider, T., Lan, S., Stuart, A., and Teixeira, J. (2017). Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations. *Geophysical Research Letters*, 44(24):12–396.

Shepard, D. (1968). A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM National Conference*, ACM '68, page 517–524, New York, NY, USA. Association for Computing Machinery.

Shi, X., Gao, Z., Lausen, L., Wang, H., Yeung, D.-Y., Wong, W.-k., and Woo, W.-c. (2017). Deep learning for precipitation nowcasting: A benchmark and a new model. In *Advances in neural information processing systems*, pages 5617–5627.

Simmons, A. J. and Hollingsworth, A. (2002). Some aspects of the improvement in skill of numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 128(580):647–677.

Spirtes, P. and Glymour, C. (1991). An algorithm for fast recovery of sparse causal graphs. *Social science computer review*, 9(1):62–72.

Spirtes, P., Glymour, C. N., and Scheines, R. (2000). *Causation, prediction, and search.* MIT press.

Srinivasan, V., Khim, J., Banerjee, A., and Ravikumar, P. (2021). Subseasonal climate prediction in the western us using bayesian spatial models. *Conference on Uncertainty in Artificial Intelligence (UAI).*

Srivastava, N., Mansimov, E., and Salakhudinov, R. (2015). Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852.

Stan, C., Straus, D. M., Frederiksen, J. S., Lin, H., Maloney, E. D., and Schumacher, C. (2017). Review of tropical-extratropical teleconnections on intraseasonal time scales. *Reviews of Geophysics*, 55(4):902–937.

Steinhaeuser, K., Chawla, N., and Ganguly, A. (2011a). Comparing predictive power in climate data: Clustering matters. *Advances in Spatial and Temporal Databases,*, pages 39–55.

Steinhaeuser, K., Chawla, N. V., and Ganguly, A. R. (2011b). Complex networks as a unified framework for descriptive analysis and predictive modeling in climate science. *Statistical Analysis and Data Mining,*, 4(5):497–511.

Stevens, A., Willett, R., Mamalakis, A., Foufoula-Georgiou, E., Tejedor, A., Randerson, J. T., Smyth, P., and Wright, S. (2021). Graph-guided regularized regression of pacific ocean climate variables to increase predictive skill of southwestern us winter precipitation. *Journal of Climate*, 34(2):737–754.

Stocker, T. F., Qin, D., Plattner, G.-K., Tignor, M., Allen, S. K., Boschung, J., Nauels, A., Xia, Y., Bex, V., and Midgley, P. M. (2013). Climate change 2013: The physical science basis. *Intergovernmental Panel on Climate Change, Working Group I Contribution to the IPCC Fifth Assessment Report (AR5)(Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA.)*.

Strobach, E. and Bel, G. (2016). Decadal climate predictions using sequential learning algorithms. *Journal of Climate*, 29(10):3787–3809.

Sundermeyer, M., Schlüter, R., and Ney, H. (2012). Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Taylor, K. E., Stouffer, R. J., and Meehl, G. A. (2012). An overview of cmip5 and the experiment design. *Bulletin of the American Meteorological Society,*, 93(4):485–498.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society,*, pages 267–288.

Trewin, B., Baddour, O., Organization, W. M., Kontongomde, H., Data, W. C., and Programme, M. (2007). *The Role of Climatological Normals in a Changing Climate*. WCDMP (Series). World Meteorological Organization.

Tucker, G., Mnih, A., Maddison, C. J., Lawson, J., and Sohl-Dickstein, J. (2017). Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. *Advances in Neural Information Processing Systems*, 30.

Uccellini, L. W. and Jacobs, N. G. (2018). *Subseasonal and Seasonal Forecasting Innovation: Plans for the Twenty-First Century*. National Weather Service (U.S.).

USBR and NOAA (2019). Forecast Rodeo II. `https://www.challenge.gov/challenge/`

`rodeo-ii-sub-seasonal-climate-forecasting/`.

Vahdat, A., Andriyash, E., and Macready, W. (2018). Dvae#: Discrete variational autoencoders with relaxed boltzmann priors. *Advances in Neural Information Processing Systems*, 31.

Vahdat, A., Andriyash, E., and Macready, W. (2020). Undirected graphical models as approximate posteriors. In *International Conference on Machine Learning*, pages 9680–9689. PMLR.

Vahdat, A. and Kautz, J. (2020). Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679.

Van den Dool, H. (2007). *Empirical methods in short-term climate prediction.* Oxford University Press.

Van den Dool, H., Saha, S., and Johansson, A. (2000). Empirical orthogonal teleconnections. *Journal of Climate*, 13(8):1421–1435.

Vandal, T., Kodra, E., Ganguly, S., Michaelis, A., Nemani, R., and Ganguly, A. R. (2017). Deepsd: Generating high resolution climate change projections through single image super-resolution. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 1663–1672.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., and Saenko, K. (2015). Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542.

Vichi, M., Manzini, E., Fogli, P. G., Alessandri, A., Patara, L., Scoccimarro, E., Masina, S., and Navarra, A. (2011). Global and regional ocean carbon uptake and climate change: sensitivity to a substantial mitigation scenario. *Climate dynamics,*, 37(9-10):1929–1947.

Villani, C. (2009). *Optimal transport: old and new*, volume 338. Springer.

Vitart, F. (2004). Monthly forecasting at ecmwf. *Monthly Weather Review*, 132(12):2761–2779.

Vitart, F. (2014). Evolution of ecmwf sub-seasonal forecast skill scores. *Quarterly Journal of the Royal Meteorological Society*, 140(683):1889–1899.

Vitart, F., Ardilouze, C., Bonet, A., Brookshaw, A., Chen, M., Codorean, C., Déqué, M., Ferranti, L., Fucile, E., Fuentes, M., et al. (2017). The subseasonal to seasonal (s2s) prediction project database. *Bulletin of the American Meteorological Society*, 98(1):163–173.

Vitart, F., Robertson, A. W., and Anderson, D. L. (2012). Subseasonal to seasonal prediction project: Bridging the gap between weather and climate. *Bulletin of the World*

Meteorological Organization, 61(23).

Volodin, E., Dianskii, N., and Gusev, A. (2010). Simulating present-day climate with the inmcm4.0 coupled model of the atmospheric and oceanic general circulations. *Atmospheric and Oceanic Physics,*, 46(4):414–431.

Von Storch, H. and Zwiers, F. W. (2001). *Statistical analysis in climate research.* Cambridge university press.

Waliser, D. (2005). *Predictability and forecasting*, pages 389–423. Springer Berlin Heidelberg, Berlin, Heidelberg.

Walter, C., McBratney, A. B., Douaoui, A., and Minasny, B. (2001). Spatial prediction of topsoil salinity in the chelif valley, algeria, using local ordinary kriging with local variograms versus whole-area variogram. *Soil Research*, 39(2):259–272.

Wang, C., Jia, Z., Yin, Z., Liu, F., Lu, G., and Zheng, J. (2021). Improving the accuracy of subseasonal forecasting of china precipitation with a machine learning approach. front. *Earth Sci*, 9:659310.

Wang, H. and Banerjee, A. (2014). Bregman alternating direction method of multipliers. In *Advances in Neural Information Processing Systems.*

Wang, H., Banerjee, A., Hsieh, C.-J., Ravikumar, P. K., and Dhillon, I. S. (2013). Large scale distributed sparse precision estimation. In *Advances in Neural Information Processing Systems*, pages 584–592.

Wasserman, L. (2013). *All of statistics: a concise course in statistical inference.* Springer Science & Business Media.

Watanabe, M., Suzuki, T., Oishi, R., Komuro, Y., Watanabe, S., Emori, S., Takemura, T., Chikira, M., Ogura, T., Sekiguchi, M., et al. (2010). Improved climate simulation by miroc5: mean states, variability, and climate sensitivity. *Journal of Climate,*, 23(23):6312–6335.

Weigel, A. P., Baggenstos, D., Liniger, M. A., Vitart, F., and Appenzeller, C. (2008). Probabilistic verification of monthly temperature forecasts. *Monthly Weather Review*, 136(12):5162–5182.

Weisberg, S. (2005). *Applied linear regression*, volume 528. John Wiley & Sons.

Weng, L. (2019). From gan to wgan. *arXiv preprint arXiv:1904.08994.*

Weyn, J. A., Durran, D. R., Caruana, R., and Cresswell-Clay, N. (2021). Sub-seasonal forecasting with a large ensemble of deep-learning weather prediction models. *arXiv preprint arXiv:2102.05107.*

Wheeler, M. C. and Hendon, H. H. (2004). An all-season real-time multivariate mjo index: Development of an index for monitoring and prediction. *Monthly Weather Review*, 132(8):1917–1932.

White, C. J., Carlsen, H., Robertson, A. W., Klein, R. J., Lazo, J. K., Kumar, A., Vitart, F., Coughlan de Perez, E., Ray, A. J., Murray, V., et al. (2017). Potential applications of subseasonal-to-seasonal (s2s) predictions. *Meteorological applications*, 24(3):315–325.

Wikipedia (2019). January–February 2019 North American cold wave. `https://en.wikipedia.org/wiki/January-February_2019_North_American_cold_wave`.

Wilks, D. S. (2011). *Statistical methods in the atmospheric sciences*, volume 100. Academic press.

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

Xue, L., Zou, H., et al. (2012). Regularized rank-based estimation of high-dimensional nonparanormal graphical models. *The Annals of Statistics*, 40(5):2541–2571.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in neural information processing systems (NIPS)*, pages 3320–3328.

Zhang, T., Hoell, A., Perlwitz, J., Eischeid, J., Murray, D., Hoerling, M., and Hamill, T. M. (2019). Towards probabilistic multivariate enso monitoring. *Geophysical Research Letters*, 46(17-18):10532–10540.

Zimmerman, B. G., Vimont, D. J., and Block, P. J. (2016). Utilizing the state of enso as a means for season-ahead predictor selection. *Water resources research*, 52(5):3761–3774.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association,*, 101(476):1418–1429.

Part V
Appendix

Appendix A

Interpretable Predictive Modeling for Climate Variables with Weighted Lasso

A.1 The Restricted Error Set

To prove Theorem 1, Lemma 1 is proposed to characterize the set to which the error vector $\Delta = \hat{\theta} - \theta^*$ belongs.

Lemma 1. *Assuming*

$$\lambda \geq O(\max\{\frac{\sqrt{m}}{\sqrt{n}\|w^{\uparrow}_{1:m}\|_2}, \frac{\sqrt{\log p}}{\sqrt{n}\tilde{w}_{min}}\}) \ , \tag{A.1}$$

where $\tilde{w}_{\min}$ is the minimum element in $\mathcal{M}^{\perp}(w^{\uparrow})$, the error set is

$$E_r = \{\Delta \in \mathbb{R}^p | R(\mathcal{M}^{\perp}(\Delta)) \leq \beta \|w^{\uparrow}_{1:m}\|_2 \|\mathcal{M}(\Delta)\|_2\} \ , \tag{A.2}$$

where $\beta > 1$ is a constant and $R(\theta) = \sum_{i=1}^{p} w_i \, |\theta_i|$ is the regularizer of weighted Lasso.

Proof. By the optimality of $\hat{\theta} = \theta^* + \Delta$, we have

$$\mathcal{L}(\theta^* + \Delta) + \lambda R(\theta^* + \Delta) - \{\mathcal{L}(\theta^*) + \lambda R(\theta^*)\} \leq 0 \ , \tag{A.3}$$

where $\mathcal{L}$ is loss function.

Further, since $\mathcal{L}(\theta) = \frac{1}{2n}\|y - X\theta\|_2^2$, we have

$$\begin{aligned}
&\mathcal{L}(\theta^* + \Delta) - \mathcal{L}(\theta^*) \geq -\frac{1}{n}\langle X^T\epsilon, \Delta\rangle \\
&\geq -\frac{1}{n}\langle \mathcal{M}(X^T\epsilon), \mathcal{M}(\Delta)\rangle - \frac{1}{n}\langle \mathcal{M}^\perp(X^T\epsilon), \mathcal{M}^\perp(\Delta)\rangle \\
&\geq -\frac{\|\epsilon\|_2}{n}\langle \mathcal{M}(X^Tu), \mathcal{M}(\Delta)\rangle - \frac{\|\epsilon\|_2}{n}\langle \mathcal{M}^\perp(X^Tu), \mathcal{M}^\perp(\Delta)\rangle \\
&\geq -\frac{c}{\sqrt{n}}\langle \mathcal{M}(X^Tu), \mathcal{M}(\Delta)\rangle - \frac{c}{\sqrt{n}}\langle \mathcal{M}^\perp(X^Tu), \mathcal{M}^\perp(\Delta)\rangle \ ,
\end{aligned} \tag{A.4}$$

where $c > 0$ is a constant, $\|\epsilon\|_2 = O(\sqrt{n})$ with high probability since $\epsilon \in R^n$ has i.i.d. centered unit-variance sub-Gaussian entries and $u \in S^{n-1}$ is unit vector on S^{n-1}. By generalized Holder's inequality, we have

$$\mathcal{L}(\theta^* + \Delta) - \mathcal{L}(\theta^*) \geq -\frac{c}{\sqrt{n}}\|\mathcal{M}(X^Tu)\|_2\|\mathcal{M}(\Delta)\|_2 - \frac{c}{\sqrt{n}}\langle \mathcal{M}^\perp(X^Tu), \mathcal{M}^\perp(\Delta)\rangle \tag{A.5}$$

Considering the first term in the RHS of (A.5), if X has isotropic sub-Gaussian rows, then X^Tu is a sub-Gaussian vector with covariance matrix as an identity matrix $\mathbb{I}_{p\times p}$. Therefore, we have $\|\mathcal{M}(X^Tu)\|_2 = O(\sqrt{m})$. Then, for the second term in the RHS of (A.5),

$$\begin{aligned}
&\frac{c}{\sqrt{n}}\langle \mathcal{M}^\perp(X^Tu), \mathcal{M}^\perp(\Delta)\rangle \\
&= \frac{cR(\mathcal{M}^\perp(\Delta))}{\sqrt{n}}\langle \mathcal{M}^\perp(g), \frac{\mathcal{M}^\perp(\Delta)}{R(\mathcal{M}^\perp(\Delta)}\rangle \\
&= \frac{cR(\mathcal{M}^\perp(\Delta))}{\sqrt{n}}G(\Omega_2) \ ,
\end{aligned} \tag{A.6}$$

where $\Omega_2 = \{u \in \mathbb{R}^{(p-m)} | R(u) \leq 1\}$. $G(A)$ is Gaussian width Gordon (1988) of set A. It is defined as $G(A) = E_g[\sup_{t\in A}\langle g, t_i\rangle]$, where the expectation is over $g \sim N(0, \mathbb{I}_{p\times p})$.

Note $R(u)$ norm can be viewed as the atomic norm induced by set $\mathcal{A}_{\text{awl}} = \cup_{1\leq i\leq q}\mathcal{A}_i = \cup_{1\leq i\leq q}\{\pm\frac{\mathbf{e}_i}{w_i}\}$, where $q = p - m$, $\{\mathbf{e}_i\}_{i=1}^q$ is the canonical basis of R^q and $|\mathcal{A}_{\text{awl}}|$, the cardinality of $\mathcal{A}_{\text{awl}}$, is equal to $2q$. According to Lemma 2 in Chen and Banerjee (2015), the

Gaussian width of the norm ball Ω_2 can be bounded by the atomic norm as follows

$$\begin{aligned} G(\Omega_2) &= G(\mathcal{A}_{awl}) \\ &\leq \max_{1\leq i\leq 2q} G(\mathcal{A}_i) + 2 \sup_{z\in\mathcal{A}_{awl}} \|z\|_2 \sqrt{\log(2q)} \\ &= 0 + \frac{2}{\tilde{w}_{\min}} \sqrt{\log(2q)} \\ &= O(\frac{\sqrt{\log q}}{\tilde{w}_{\min}}) = O(\frac{\sqrt{\log p}}{\tilde{w}_{\min}}) \ , \end{aligned} \tag{A.7}$$

where $\tilde{w}_{\min}$ is the minimum element in $\mathcal{M}^{\perp}(w^{\uparrow})$. Substituting $G(\Omega_2)$ from (A.7) to (A.6), we have

$$\frac{1}{\sqrt{n}}\langle \mathcal{M}^{\perp}(X^T u), \mathcal{M}^{\perp}(\Delta)\rangle \leq \frac{cR(\mathcal{M}^{\perp}(\Delta))\sqrt{\log p}}{\sqrt{n}\tilde{w}_{\min}} \ . \tag{A.8}$$

Combining (A.4), (A.5) and (A.8), we have

$$\begin{aligned} &\mathcal{L}(\theta^* + \Delta) - \mathcal{L}(\theta^*) \\ &\geq -c_1\sqrt{\frac{m}{n}}\|\mathcal{M}(\Delta)\|_2 - c_2\frac{R(\mathcal{M}^{\perp}(\Delta))\sqrt{\log p}}{\sqrt{n}\tilde{w}_{\min}} \ , \end{aligned} \tag{A.9}$$

where $c_1; c_2 > 0$ are constants. Also, since $R(\theta)$ is a decomposable norm and $\theta^* \in \mathcal{M}$, with triangle inequality, we have

$$\begin{aligned} &R(\theta^* + \Delta) - R(\theta^*) \\ &= R(\theta^* + \mathcal{M}(\Delta) + \mathcal{M}^{\perp}(\Delta)) - R(\theta^*) \\ &= R(\theta^* + \mathcal{M}(\Delta)) + R(\mathcal{M}^{\perp}(\Delta)) - R(\theta^*) \\ &\geq R(\theta^*) - R(\mathcal{M}(\Delta)) + R(\mathcal{M}^{\perp}(\Delta)) - R(\theta^*) \\ &\geq R(\mathcal{M}^{\perp}(\Delta)) - R(\mathcal{M}(\Delta)) \ , \end{aligned} \tag{A.10}$$

where $\mathcal{M}^{\perp}$ is the orthogonal subspace of $\mathcal{M}$. Combining (A.3), (A.9) and (A.10), we have

$$\mathcal{L}(\theta^* + \Delta) - \mathcal{L}(\theta^*) + \lambda(R(\mathcal{M}^{\perp}(\Delta)) - R(\mathcal{M}(\Delta))) \leq 0 \tag{A.11}$$

$$\begin{aligned} \Rightarrow -c_1\sqrt{\frac{m}{n}}\|\mathcal{M}(\Delta)\|_2 - c_2\frac{R(\mathcal{M}^{\perp}(\Delta))\sqrt{\log p}}{\sqrt{n}\tilde{w}_{\min}} \\ \leq \lambda(R(\mathcal{M}(\Delta)) - R(\mathcal{M}^{\perp}(\Delta))) \end{aligned} \tag{A.12}$$

$$\begin{aligned}\Rightarrow -c_1\sqrt{\frac{m}{n}}\|\mathcal{M}(\Delta)\|_2 - c_2\frac{R(\mathcal{M}^{\perp}(\Delta))\sqrt{\log p}}{\sqrt{n}\tilde{w}_{\min}} \\ \leq \lambda(\|w^{\uparrow}_{1:m}\|_2\|\mathcal{M}(\Delta)\|_2 - R(\mathcal{M}^{\perp}(\Delta)))\end{aligned} \tag{A.13}$$

$$\begin{aligned}\Rightarrow R(\mathcal{M}^{\perp}(\Delta))(\lambda - c_2\frac{R(\mathcal{M}^{\perp}(\Delta))\sqrt{\log p}}{\sqrt{n}\tilde{w}_{\min}}) \\ \leq (c_1\sqrt{\frac{m}{n}} + \lambda\|w^{\uparrow}_{1:m}\|_2)\|\mathcal{M}(\Delta)\|_2 \ .\end{aligned} \tag{A.14}$$

Since $\lambda \geq O(\max\{\frac{\sqrt{m}}{\sqrt{n}\|w^{\uparrow}_{1:m}\|_2}, \frac{\sqrt{\log p}}{\sqrt{n}\tilde{w}_{\min}}\})$,

$$R(\mathcal{M}^{\perp}(\Delta)) \leq \beta\|w^{\uparrow}_{1:m}\|_2\|\mathcal{M}(\Delta)\|_2. \tag{A.15}$$

Thus, we complete the proof. □

A.2 Estimation Error of Weighted Lasso

Below we provide the proof of Theorem 1.

Proof. The weigthed Lasso estimator is of the form

$$\hat{\theta} = \text{argmin}_{\theta\in R^p}\ \frac{1}{2n}\|y - X\theta\|^2 + \lambda\sum_{i=1}^{p} w_i|\theta_i| \ , \tag{A.16}$$

where $\hat{\theta}$ is the estimated parameter vector and λ is the regularization parameter.

For λ in Lemma 1, the error vector $\Delta = \hat{\theta} - \theta^*$ belongs to the restricted error set

$$E_r = \{\Delta \in \mathbb{R}^p | R(\mathcal{M}^{\perp}(\Delta)) \leq \beta\|w^{\uparrow}_{1:m}\|_2\|\mathcal{M}(\Delta)\|_2\}. \tag{A.17}$$

Furthermore, the loss function $\mathcal{L}(\theta) = \frac{1}{2n}\|y - X\theta\|_2^2$ is assumed to satisfy the restricted eigenvalue (RE) condition Bickel et al. (2009). Define $\delta\mathcal{L}(\Delta, \theta^*)$ as

$$\delta\mathcal{L}(\Delta, \theta^*) \triangleq \mathcal{L}(\theta^* + \Delta) - \mathcal{L}(\theta^*) - \langle\nabla\mathcal{L}(\theta^*), \Delta\rangle \ , \tag{A.18}$$

then there exists a suitable constant $\kappa > 0$ such that, with high probability,

$$\delta\mathcal{L}(\Delta, \theta^*) = \frac{1}{2n}\|X\Delta\|_2^2 \geq \kappa\|\Delta\|_2^2, \ \ \forall\Delta \in E_r. \tag{A.19}$$

From (A.4) and (A.9) in Lemma 1, we have

$$\begin{aligned}\langle \nabla \mathcal{L}(\theta^*), \Delta \rangle &= -\frac{1}{n}\langle X^T\epsilon, \Delta\rangle \\ &\geq -c_1\sqrt{\frac{m}{n}}\|\mathcal{M}(\Delta)\|_2 - c_2\frac{R(\mathcal{M}^{\perp}(\Delta))\sqrt{\log p}}{\sqrt{n}\tilde{w}_{\min}}\end{aligned} \tag{A.20}$$

Further, from triangle inequality, we have

$$R(\theta^* + \Delta) - R(\theta^*) \geq -R(\Delta)\ . \tag{A.21}$$

Deriving from (A.18) and (A.19), we have

$$\begin{aligned}\mathcal{L}(\theta^* + \Delta) - \mathcal{L}(\theta^*) &= \langle \nabla \mathcal{L}(\theta^*), \Delta\rangle + \frac{1}{2n}\|X\Delta\|_2^2 \\ &\geq \langle \nabla \mathcal{L}(\theta^*), \Delta\rangle + \kappa\|\Delta\|_2^2\ .\end{aligned} \tag{A.22}$$

Adding (A.21) and (A.22), we have

$$\begin{aligned}\mathcal{F}(\Delta) &= \mathcal{L}(\theta^* + \Delta) - \mathcal{L}(\theta^*) + \lambda(R(\theta^* + \Delta) - R(\theta^*)) \\ &\geq - c_1\sqrt{\frac{m}{n}}\|\mathcal{M}(\Delta)\|_2 - c_2\frac{R(\mathcal{M}^{\perp}(\Delta))\sqrt{\log p}}{\sqrt{n}\tilde{w}_{\min}} \\ &\quad + \kappa\|\Delta\|_2^2 - \lambda R(\Delta).\end{aligned} \tag{A.23}$$

Since $\mathcal{F}(\Delta) \leq 0$, we have

$$\kappa\|\Delta\|_2^2 \leq c_1\sqrt{\frac{m}{n}}\|\mathcal{M}(\Delta)\|_2 + c_2\frac{R(\mathcal{M}^{\perp}(\Delta))\sqrt{\log p}}{\sqrt{n}\tilde{w}_{\min}} + \lambda R(\Delta). \tag{A.24}$$

Also, from Lemma 1, we have

$$\begin{aligned}\frac{R(\Delta)}{\|\Delta\|_2} &\leq \frac{R(\mathcal{M}(\Delta)) + R(\mathcal{M}^{\perp}(\Delta))}{\|\mathcal{M}(\Delta)\|_2} \\ &\leq \frac{(\beta+1)\|w^{\uparrow}_{1:m}\|_2\|\mathcal{M}(\Delta)\|_2}{\|\mathcal{M}(\Delta)\|_2} \\ &\leq (\beta+1)\|w^{\uparrow}_{1:m}\|_2\ ,\end{aligned} \tag{A.25}$$

where β is a constant.

Then, dividing (A.24) by $\kappa\|\Delta\|_2$, we get

$$\begin{aligned}\|\Delta\|_2 &\leq c_1 \frac{\sqrt{m}\|\mathcal{M}(\Delta)\|_2}{\kappa\sqrt{n}\|\Delta\|_2} + c_2 \frac{R(\mathcal{M}^{\perp}(\Delta))\sqrt{\log p}}{\kappa\sqrt{n}\tilde{w}_{\min}\|\Delta\|_2} + \frac{\lambda R(\Delta)}{\kappa\|\Delta\|_2} \\ &\leq c_1 \frac{\sqrt{m}}{\kappa\sqrt{n}} + c_2 \frac{\|w^{\uparrow}_{1:m}\|_2\sqrt{\log p}}{\kappa\sqrt{n}\tilde{w}_{\min}} + c_3 \frac{\lambda\|w^{\uparrow}_{1:m}\|_2}{\kappa} \ ,\end{aligned} \tag{A.26}$$

where $c_1, c_2, c_3 > 0$ are constants.

Then, by substituting value for λ from Lemma 1, we obtain

$$\|\Delta\|_2 \leq \frac{c}{\sqrt{n}\kappa}\left(\sqrt{m} + \frac{\|w^{\uparrow}_{1:m}\|_2\sqrt{\log p}}{\tilde{w}_{\min}}\right) \ , \tag{A.27}$$

where $c > 0$ is a constant. The proof is completed.

□

Appendix B

Sub-Seasonal Climate Forecasting via Machine Learning: Challenges, Analysis, and Advances

B.1 Difficulty of the Problem

B.1.1 Dependence Between Historical Data and Forecasting Target

In section 4.3, the dependence between the most recent historical data (the residualized average temperature of week -2 & -1) and the forecasting target (the residualized average temperature of week 3 & 4) is measured by maximum information coefficient (MIC). Here we show the results measured by Pearson correlation coefficient Wasserman (2013), and Spearman's rank correlation coefficient Wasserman (2013) (Figure B.1). Small values ($\leq$0.2) of Pearson correlation and Spearman's rank correlation at a majority of locations, which verify that there is little information shared between the most recent date and the forecasting target, once again, demonstrate how difficult SSF is.

B.1.2 Relative $\mathbf{R^2}$

In Chapter 4, we introduce cosine similarity, which is widely used in weather prediction evaluation, as an evaluation metric. Here we formally define the other evaluation metric, namely relative R^2 as

$$\text{Relative } R^2 = 1 - \text{Relative MSE} = 1 - \frac{\sum_{i=1}^{n}(\mathbf{y}_i^* - \hat{\mathbf{y}}_i)^2}{\sum_{i=1}^{n}(\mathbf{y}_i^* - \bar{\mathbf{y}}_{\text{train}})^2} \ , \tag{B.1}$$

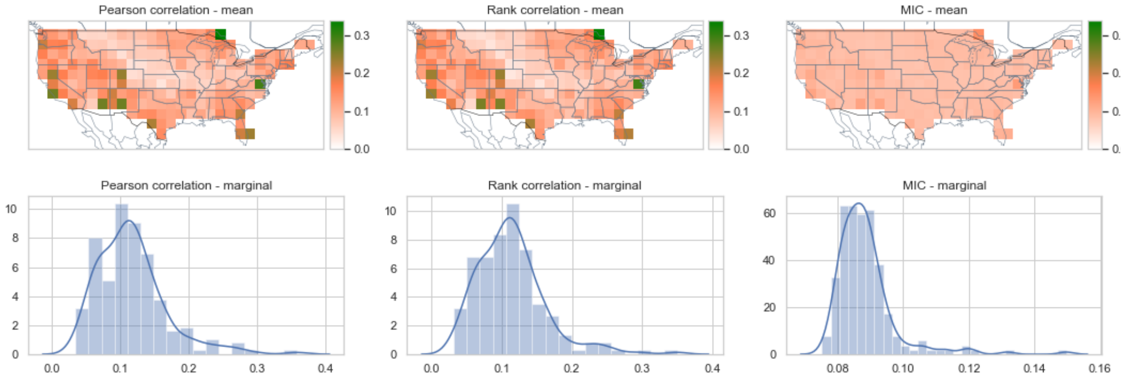


Figure B.1: Pearson correlation, Spearman's rank correlation and MIC between 2m temperature of week -2 & -1 and week 3 & 4. Small values ($\leq$0.2) of Pearson correlation and Spearman's rank correlation at a majority of locations verify the fact, as we illustrate using MIC, that there is little information shared between the most recent date and the forecasting target.

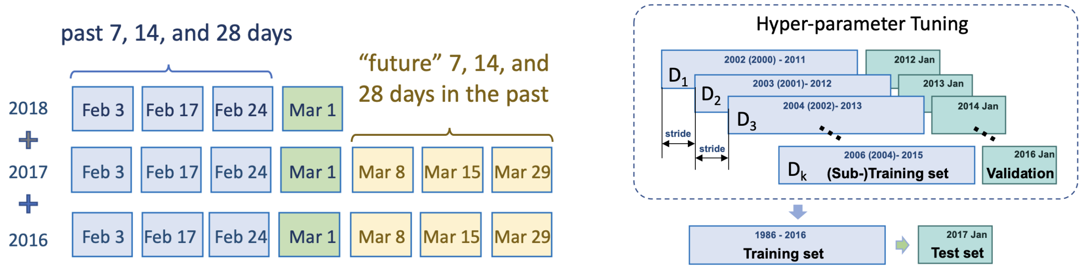
where $\hat{\mathbf{y}}$ denotes a vector of predicted values, and $\mathbf{y}^*$ be the corresponding ground truth. We use relative R^2 to evaluate the relative predictive skill of a given prediction $\hat{y}$ compared to the best constant predictor $\bar{y}_{\text{train}}$, the long-term average of target variable at each date and each target location computed from training set. A model which achieves a positive relative R^2 is, at least, able to predict the sign of y^* accurately. The results of temporal and spatial relative R^2 over the US mainland of ML models discussed in section 4.4 are shown in Table B.1 and Figure B.3 respectively.

B.2 Data and Experimental Setup

B.2.1 Data Sources

The data described in Table 4.1 were downloaded from the following sources:

- Temperature (tmp2m): `https://www.esrl.noaa.gov/psd/data/gridded/data.cpc.globaltemp.html`
- Soil moisture (sm): `https://www.esrl.noaa.gov/psd/data/gridded/data.cpcsoil.html`
- Sea surface temperature (sst): `https://www.ncdc.noaa.gov/oisst`
- Relative humidity (rhum), sea level pressure (slp), and geopotential height (hgt): `ftp://ftp.cdc.noaa.gov/Datasets/ncep.reanalysis/surface/`



(a) Sequential feature set for Mar 1, 2018

(b) Evaluation pipeline

Figure B.2: (a) Sequential feature set: to construct feature set at Mar. 1, 2018, we concatenate covariates from Mar. 1 in 2018, 2017, and 2016, their corresponding 7^{th}, 14^{th}, and 28^{th} days in the past, and 7^{th}, 14^{th}, and 28^{th} "future" days in 2017 and 2016. (b) Evaluation pipeline: to test SSF in Jan 2017, the training set covers historical 30 year ends at Dec 4, 2016 (the last available date). 5 validation sets include dates from each Jan between 2012 to 2016, with the corresponding training sets generated by applying a moving window of 10 years and a stride of 365 days on data start from 2000.

- Multivariate ENSO index (MEI): `https://psl.noaa.gov/enso/mei/`
- Niño indices: `https://www.cpc.ncep.noaa.gov/data/indices/wksst8110.for`
- North Atlantic Oscillation (NAO) index: `ftp://ftp.cpc.ncep.noaa.gov/cwlinks/norm.daily.nao.index.b500101.current.ascii`
- Madden Julian Oscillation (MJO) phase & amplitude: `http://www.bom.gov.au/climate/mjo/graphics/rmm.74toRealtime.txt`

B.2.2 PCA Prepossessing

As mentioned in section 4.5, one way for feature extraction is to apply PCA to spatial-temporal variables. To do so, let's consider sst of Pacific ocean as an example. Daily sst of Pacific ocean is originally stored in a matrix, of which each element represents the sea surface temperature at each grid point of Pacific ocean. The covariance matrix can be computed by flattening each matrix into a 1-D vector, viewing each element in the matrix as a feature and each date as one observation. Such covariance matrix captures spatial connection among grid points of Pacific ocean. By considering all dates from 1986 to 2016, we can extract the top 10 principal components (PCs) as features based on PC loadings computed from the corresponding covariance.

B.2.3 Feature Set Construction

To better utilize historical information, we construct a sequential feature set by including not only covariates of the target date, but also covariates of the 7^{th}, 14^{th}, and 28^{th} day previous from the target date, as well as the day of the year of the target date in the past 2 years and both the historical past and future dates around the day of the year of the target date in the past 2 years. Such selection of historical dates mainly bases on the temporal correlation. Figure B.2(a) provides a detailed example on how to construct feature set for Mar 1, 2018: we concatenate covariates from Mar. 1 in 2018, 2017, and 2016, their corresponding 7^{th}, 14^{th}, and 28^{th} days in the past, and 7^{th}, 14^{th}, and 28^{th} "future" days in 2017 and 2016. In total, we include $H = 18$ historical days in our feature set for each date.

B.2.4 Evaluation Pipeline

Predictive models are created independently for each month in 2017 and 2018. To mimic a live forecasting system, we generate 105 test dates during 2017-2018, one for each week, and group them into 24 test sets by their month of the year. Given a test set, our evaluation pipeline consists of two parts (Figure B.2(b)):

- "5-fold" training-validation pairs for hyper-parameter tuning, based on a "sliding-window" strategy designed for time-series data. Each validation set uses the data from the same month of the year as the test set. For instance, if the test set is Jan 2017, the corresponding 5 validation sets are Jan 2012, Jan 2013, Jan 2014, Jan 2015, and Jan 2016 respectively. Each validation set corresponds to a training set containing 10 years of data and ending 28 days before the first date in the validation set. Specifically, if the validation set starting from Jan 1, 2016, the training set is from Dec 4, 2005 to Dec 4, 2015. Such construction is equivalent to apply a sliding-window of 10-year with a stride of 365 days on data from 2002.
- The training-test pair, where the training set, including 30-year data in the past, ends 28 days before the first date in the test set. For example, to test SSF in Jan 2017, i.e., Jan 1, Jan 8, Jan 15, Jan 22, and Jan 29, the training set starts from Dec 4, 1986 and ends at Dec 4, 2016, which is the 28^{th} day before Jan 1, and the last date we have the ground truth for the target variable.

Table B.1: Comparison of relative R^2 of tmp2m forecasting for test sets over 2017-2018. A positive relative R^2 indicates a model predicting the sign of the target variable correctly. XGBoost achieves the highest relative R^2.

Model	Mean(se)	Median (se)	0.25 quantile (se)	0.75 quantile (se)
Temporally Global Set				
XGBoost - one day	**0.0760(0.03)**	**0.0974(0.03)**	**-0.0449(0.03)**	**0.2434(0.03)**
Lasso - one day	0.0552(0.02)	0.0321(0.02)	-0.0309(0.01)	0.1295(0.02)
Encoder (LSTM)-Decoder (FNN)	**-0.0353 (0.05)**	**0.0596(0.05)**	**-0.2409 (0.06)**	**0.2426 (0.05)**
FNN	-0.5777(0.29)	-0.0183(0.15)	-0.0794(0.13)	0.0213(0.13)
CNN	-0.0564(0.03)	0.0284(0.02)	-0.0266(0.02)	0.0570(0.02)
CNN-LSTM	-0.1164(0.05)	0.0263(0.03)	-0.0862(0.03)	0.0698(0.03)
LS with NAO & all nino - daily	**0.0418(0.01)**	**0.0535(0.01)**	**-0.0078(0.01)**	**0.0949(0.01)**
Damped persistence	0.0266(0.01)	0.0414(0.02)	-0.0542(0.02)	0.1354(0.02)
MultiLLR	-0.0571 (0.02)	0.0034 (0.02)	-0.1156 (0.03)	0.0797 (0.02)
AutoKNN	0.0181 (0.01)	0.0260 (0.02)	-0.0531 (0.02)	0.1041 (0.01)
Temporally Local Set				
XGBoost - one day	-0.0337(0.03)	0.0396(0.03)	-0.1310(0.04)	0.1873(0.03)
Lasso - one day	-0.0028(0.02)	0.0327(0.02)	-0.0613(0.02)	0.0996(0.02)
Encoder (LSTM)-Decoder (FNN)	-0.2333 (0.06)	-0.1116 (0.06)	-0.4694 (0.09)	0.1808 (0.06)

B.2.5 Hyperparameters Tuning

We largely use standard hyperparameters for each of the methods. To be specific, we perform hyperparameter tuning for the models and their hyperparameter(s) described below:

- Lasso: the penalty parameter λ.
- XGBoost: learning rate, number of gradient boosted trees, maximum depth of a tree, and other parameters related to sub-sampling.
- Encoder-Decoder style models: learning rate, number of hidden layers (depth), and number of hidden nodes (width).
- CNN-based models: learning rate, kernel size, stride, number of fully connected layers, and number of hidden nodes (width).

The optimal hyperparameters are selected based on the average performance on validation sets.

B.3 Additional Results

B.3.1 Temporal and Spatial Results of Relative $\mathbf{R^2}$

Table B.1 lists the mean, the median, the 0.25 quantile, the 0.75 quantile, and their corresponding standard errors of relative R^2 for all models. A positive relative R^2 indicates a

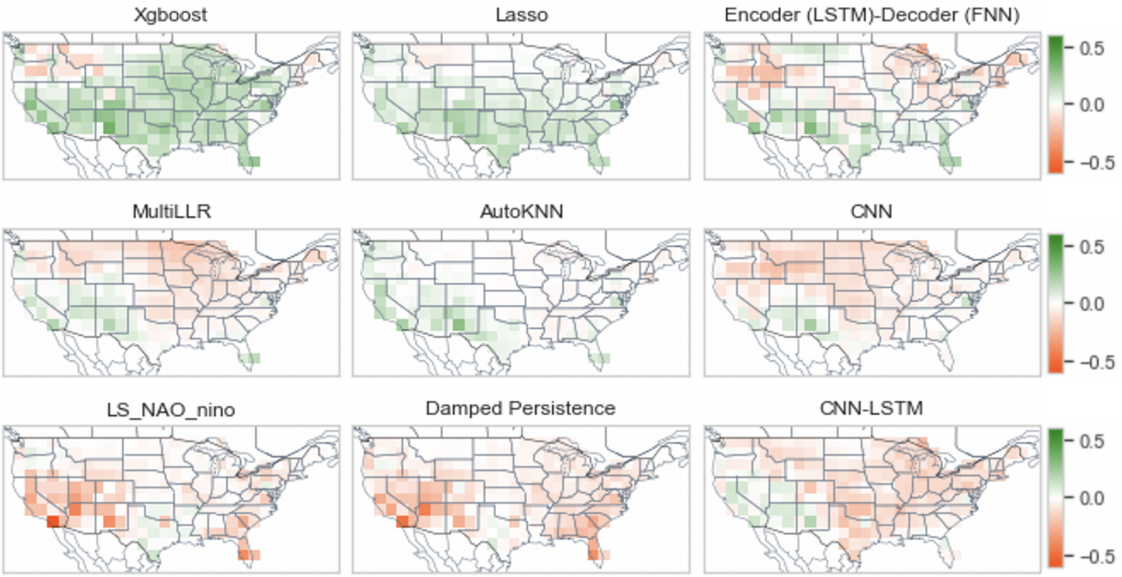
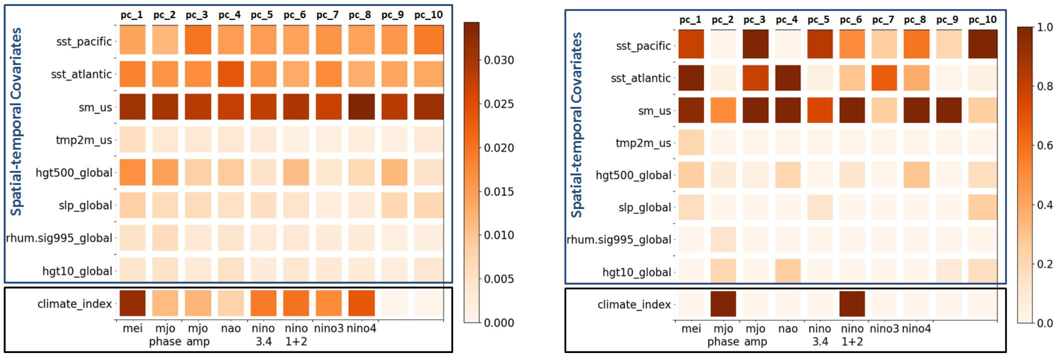


Figure B.3: Temporal relative R^2 over the US mainland of ML models discussed in section 4.4 for temperature prediction over 2017-2018. Large positive values (green) closer to 1 indicates better predictive skills.

model can at least predict the sign of the target variable correctly. Again, XGBoost achieves the highest predictive skill in terms of both the mean and the median, demonstrating its predictive power. Linear regression, like Lasso, with a proper feature set has good predictive performance. Both XGBoost and Lasso have larger positive relative R^2 in terms of the mean, and can still outperform climatology and two climate baseline models, i.e., LS with NAO & Niño, and damped persistence. Even though Encoder (LSTM)-Decoder (FNN) has a slightly negative mean relative R^2, it has the second largest median and 0.75 quantile among all models, showing its potential for further improvement.

Figure B.3 shows the spatial relative R^2 of all methods. XGBoost and Lasso are able to achieve positive relative R^2 for most of the target locations. Encoder (LSTM)-Decoder (FNN) shows better predictive skill over the southern US compared to other regions. MultiLLR and AutoKNN manages to obtain non-negative relative R^2 for the coastal area in the western US but their predictive performance drops in the rest of locations. All other baseline methods struggle to reach positive relative R^2 for most of the target locations.



(a) XGBoost

(b) Lasso

Figure B.4: Feature importance scores computed from (a) XGBoost and (b) Lasso. Darker color means a covariate is of the higher importance. The first 8 rows contains the top 10 principal components (PCs) extracted from 8 spatial-temporal covariates respectively, and the last row includes all the temporal indices. Land component, e.g., soil moisture (3^{rd} row from the top) and ocean components, e.g., sst (Pacific and Atlantic) and some climate indices are the most commonly selected covariates.

B.3.2 Analysis on Feature Importance

Besides SHAP values, we also study which covariate(s) are important, considered by ML models, based on the feature importance score. In particular, we compute the feature importance score from 2 ML models, XGBoost and Lasso (Figure B.4). For XGBoost, the importance score is computed using the average information gain across all tree nodes a feature/covariate splits, while for Lasso, we simply count the non-zero coefficients of each model. The reported feature importance score is the average over 24 models (one per month in 2017-2018). What we observe, based on feature importance scores, once again verifies our observations in Chapter 4: ML models pick up ocean-based covariates, some land-based covariates, and almost entirely ignore the atmosphere-related covariates.

To emphasis the importance of the land-based covariates, e.g., soil moisture and the ocean-based covariates, e.g., NAO and Niño indices, we compare the prediction performance among (1) the model trained with all covariates, (2) the model trained without soil moisture, and (3) the model trained without NAO and Niño indices (Table B.2 and Table B.3). Most models experience a performance deterioration when we exclude certain “important” covariates.

Table B.2: Comparison of cosine similarity of tmp2m forecasting for test sets over 2017-2018 using different feature set. Excluding soil moisture or climate indices (NAO & Niño) leads to a deterioration in the predictive performance.

Model	Mean(se)	Median (se)	0.25 quantile (se)	0.75 quantile (se)
XGBoost - one day	**0.3044(0.03)**	**0.3447(0.05)**	**0.0252(0.05)**	**0.5905(0.04)**
XGBoost - one day (w/o soil moisture)	0.2685(0.03)	0.2797(0.05)	0.0703(0.04)	0.5492(0.05)
XGBoost - one day (w/o nao & all nino)	0.2081(0.03)	0.1640(0.05)	-0.0588(0.04)	0.5246(0.05)
Lasso - one day	0.2499(0.04)	0.2554(0.06)	-0.0224(0.05)	0.5604(0.06)
Lasso - one day (w/o soil moisture)	0.2638(0.04)	0.2912(0.05)	0.0032(0.06)	0.5655(0.05)
Lasso - one day (w/o nao & all nino)	0.1956(0.04)	0.2573(0.07)	-0.1657(0.06)	0.5533(0.05)
Encoder (LSTM)-Decoder (FNN)	0.2616 (0.04)	0.2995 (0.07)	-0.0719 (0.06)	0.6310 (0.05)
Encoder (LSTM)-Decoder (FNN)(w/o soil moisture)	0.2157 (0.04)	0.2909 (0.07)	-0.1106 (0.07)	0.5443 (0.07)
Encoder (LSTM)-Decoder (FNN)(w/o nao & all nino)	0.2236 (0.04)	0.2395 (0.06)	-0.1527 (0.07)	0.5989 (0.06)

Table B.3: Comparison of relative R^2 of tmp2m forecasting for test sets over 2017-2018. Excluding soil moisture or climate indices (NAO & Niño) leads to a smaller or even negative relative R^2, showing that it becomes harder for the model to predict the sign of the target variable correctly.

Model	Mean(se)	Median (se)	0.25 quantile (se)	0.75 quantile (se)
XGBoost - one day	**0.0760(0.03)**	**0.0974(0.03)**	**-0.0449(0.03)**	**0.2434(0.03)**
XGBoost - one day (w/o soil moisture)	0.0370(0.03)	0.0322(0.03)	-0.0564(0.03)	0.2225(0.03)
XGBoost - one day (w/o nao & all nino)	-0.0161(0.03)	-0.0079(0.04)	-0.1618(0.03)	0.2426(0.04)
Lasso - one day	0.0552(0.02)	0.0321(0.02)	-0.0309(0.01)	0.1295(0.02)
Lasso - one day (w/o soil moisture)	-0.0161(0.03)	-0.0079(0.04)	-0.1618(0.03)	0.2426(0.04)
Lasso - one day (w/o nao & all nino)	0.0003(0.02)	0.0457(0.02)	-0.1113(0.03)	0.1641(0.02)
Encoder (LSTM)-Decoder (FNN)	-0.0353 (0.05)	0.0596(0.05)	-0.2409 (0.06)	0.2426 (0.05)
Encoder (LSTM)-Decoder (FNN)(w/o soil moisture)	-0.1083 (0.05)	0.0314 (0.05)	-0.3022 (0.08)	0.2252 (0.05)
Encoder (LSTM)-Decoder (FNN)(w/o nao & all nino)	-0.0802 (0.04)	0.0124 (0.05)	-0.3032 (0.06)	0.2446 (0.05)

B.3.3 The Influence of Feature Sequence Length

We compare the prediction performance under 3 different settings, referred to as "one day", "four days", and "all days" respectively. For feature set construction, "one day" includes covariates at the target date only, "four days" also covers the 7^{th}, 14^{th}, and 28^{th} days previous to the target date, and "all days" uses the exact feature sequence we use for LSTM-based models. Comparison of predictive skills under each setting, measured by both cosine similarity and relative R^2, can be found in Table B.4 and Table B.5. Both XGBoost and Lasso enjoy a performance boost using "one day" values. Especially for XGBoost, the performance of "one day" is approximately 50% better than using "all days". A possible explanation for such performance degradation as we increase the feature sequence length is that both models weight covariates from different dates exactly the same without considering temporal information, thus more noise has been introduced.

Table B.4: Comparison of spatial cosine similarity for tmp2m forecasting over 2017-2018 using various length of feature sequence. Including longer historical sequence leads to a deterioration in the predictive performance of XGBoost and Lasso.

Model	Mean(se)	Median (se)	0.25 quantile (se)	0.75 quantile (se)
XGBoost - all days	0.2080(0.03)	0.1582(0.05)	-0.0466(0.05)	0.5383(0.05)
XGBoost - four days	0.2433(0.03)	0.2203(0.05)	0.0561(0.04)	0.5168(0.06)
XGBoost - one day	**0.3044(0.03)**	**0.3447(0.05)**	**0.0252(0.05)**	**0.5905(0.04)**
Lasso - all days	0.2160(0.04)	0.2258(0.07)	-0.1381(0.06)	0.5384(0.06)
Lasso - four days	0.2247(0.04)	0.1952(0.07)	0.0572(0.06)	-0.5700(0.06)
Lasso - one day	**0.2499(0.04)**	**0.2554(0.06)**	**-0.0224(0.05)**	**0.5604(0.06)**

Table B.5: Comparison of relative R^2 (with training set mean) for tmp2m prediction for test set over 2017-2019 using different length of feature sequence. Including longer historical sequence leads to a smaller or even negative relative R^2 for both XGBoost and Lasso.

Model	Mean(se)	Median (se)	0.25 quantile (se)	0.75 quantile (se)
XGBoost - all days	-0.0200(0.03)	-0.0010(0.04)	-0.1499(0.04)	0.2304(0.04)
XGBoost - four days	0.0242(0.03)	0.0193(0.03)	-0.0786(0.03)	0.1882(0.04)
XGBoost - one day	**0.0760(0.03)**	**0.0974(0.03)**	**-0.0449(0.03)**	**0.2434(0.03)**
Lasso - all days	-0.0167(0.03)	0.0367(0.03)	-0.0639(0.02)	0.1588(0.03)
Lasso - four days	0.0518(0.02)	0.0266(0.02)	-0.0542(0.02)	0.1653(0.03)
Lasso - one day	**0.0552(0.02)**	**0.0321(0.02)**	**-0.0309(0.01)**	**0.1295(0.02)**

B.3.4 Discussion on Deep Learning Models

Results of DL models. Table B.6 and Table B.7 compare the predictive skills of 5 DL models discussed in section 2.6, measured by both cosine similarity and relative R^2. Significant improvements can been observed as we evolve from the standard Encoder (LSTM)-Decoder (LSTM), to Encoder (LSTM)-Decoder (FNN)-last step, where "last step" indicates that FNN Decoder only uses the last step of the output sequence from LSTM Encoder, and finally to Encoder (LSTM)-Decoder (FNN) with FNN Decoder uses every step of the output sequence from LSTM Encoder.

One issue with Encoder (LSTM)-Decoder (FNN) is that the input features are shared by all target locations, which requires the model to identify the useful information for each locations without any help from the input.

Autoregressive (AR) component. Currently, the Encoder(LSTM)-Decoder(FNN) clearly considers climate covariates on a global scale, which are shared by all target locations. Nevertheless, SSF depends on not only global climate condition but also local weather change. Therefore, we seek a way to improve the model by adding an autoregressive (AR) component to capture the "local" information from historical data. We consider two variants

Table B.6: Comparison of cosine similarity of tmp2m forecasting for test sets over 2017-2018 using different deep learning architectures.

Model	Mean(se)	Median (se)	0.25 quantile (se)	0.75 quantile (se)
Encoder (LSTM)-Decoder (LSTM)	0.0740 (0.03)	0.0358 (0.04)	-0.1569 (0.03)	0.2584 (0.04)
Encoder (LSTM)-Decoder (FNN)-last step	0.1614 (0.05)	0.2061 (0.08)	-0.2590 (0.08)	0.5720 (0.08)
Encoder (LSTM)-Decoder (FNN)	**0.2616 (0.04)**	**0.2995 (0.07)**	**-0.0719 (0.06)**	**0.6310 (0.05)**
Encoder (LSTM)-Decoder (FNN)+AR	0.1733 (0.04)	0.1922 (0.06)	-0.0863 (0.07)	0.5225 (0.06)
Encoder (LSTM)-Decoder (FNN)+AR (CI)	0.1852 (0.04)	0.1986 (0.05)	-0.0838 (0.06)	0.5164 (0.05)

Table B.7: Comparison of relative R^2 of tmp2m forecasting for test sets over 2017-2018. A positive relative R^2 indicates a model predicting the sign of the target variable correctly.

Model	Mean(se)	Median (se)	0.25 quantile (se)	0.75 quantile (se)
Encoder (LSTM)-Decoder (LSTM)	-0.3947 (0.05)	-0.2999 (0.05)	-0.6606 (0.08)	-0.0537 (0.05)
Encoder (LSTM)-Decoder (FNN)-last step	-0.1709 (0.06)	0.0217 (0.06)	-0.4569 (0.11)	0.2278 (0.06)
Encoder (LSTM)-Decoder (FNN)	**-0.0353 (0.05)**	**0.0596(0.05)**	**-0.2409 (0.06)**	**0.2426 (0.05)**
Encoder (LSTM)-Decoder (FNN)+AR	-0.0414 (0.04)	-0.0041 (0.05)	-0.3027 (0.07)	0.2309 (0.05)
Encoder (LSTM)-Decoder (FNN)+AR (CI)	-0.0563 (0.03)	-0.0380 (0.05)	-0.2365 (0.05)	0.1951 (0.04)

of Encoder (LSTM)-Decoder (FNN). The first variant contains an AR component with the input as historical temperature at each target location, denoted as Encoder (LSTM)-Decoder (FNN)+AR. The second one includes both historical temperature and historical temporal climate variables, i.e., climate indices, as input features, denoted as Encoder (LSTM)-Decoder (FNN)+AR (CI). For both models, the final forecast is computed as a linear combination of the prediction from Encoder (LSTM)-Decoder (FNN) and the prediction from AR component for each location. Unexpectedly, as shown in Table B.6 and Table B.7, simply adding the AR component to our Encoder(LSTM)-Decoder(FNN) does not help the model to perform better. However, we believe there is a better way to involve local information, and such modification is a promising direction that worth investigation in the future.

Table B.8: Average spatial cosine similarity for temperature forecasting over western US from 2017 to 2018.

Model	XGBoost	Lasso	Encoder(LSTM)-Decoder(FNN)	AutoKNN	MultiLLR
2017	0.2707 (0.05)	0.3401 (0.05)	0.3067 (0.06)	0.2529 (0.05)	0.0751 (0.06)
2018	0.2997(0.05)	0.2495(0.06)	0.2618(0.06)	0.1833(0.05)	0.0761(0.06)

Table B.9: Average skills of XGBoost and Lasso using SubseasonRodeo Dataset Hwang et al. (2019) for temperature forecasting of week 3 & 4. The historical forecast periods follow the description provided in Section 5.3 of Hwang et al. (2019). The results of MultiLLR and AutoKNN are from Table 2 in Hwang et al. (2019). Top two winners at each year are highlighted.

Model	2011	2012	2013	2014	2015	2016	2017	Average
XGBoost	0.2332	0.2056	**0.1932**	0.1119	**0.4164**	0.2405	0.2962	0.2424
Lasso	**0.3178**	**0.2436**	0.1860	**0.2546**	0.3606	**0.2841**	**0.3261**	**0.2818**
MultiLLR	0.2695	0.1466	0.1031	0.1973	0.3513	0.2654	**0.3079**	0.2344
AutoKNN	**0.3664**	**0.3135**	**0.2011**	**0.2775**	**0.3885**	**0.3502**	0.2807	**0.3111**

B.3.5 Comparison on Western U.S.

Table B.8 shows the average spatial cosine similarity of 5 ML models for temperature forecasting over Western U.S. AutoKNN was run following Hwang et al. (2019) except the data are normalized by z-scoring rather than just removing the long-term mean. For 2017, the results reported on Western U.S. is similar to Hwang et al. (2019). For MultiLLR, since all spatial-temporal variables are represented by PCs, some local information for each location is lost, which may lead to a drop of forecasting performance. We need to emphasize the influence of data prepossessing and feature set construction.

B.3.6 Comparison on SubseasonalRodeo Dataset

Table B.9 compares the average skills (Eq. (1) in Hwang et al. (2019)) of XGBoost and Lasso with SOTA baselines (Table 2 in Hwang et al. (2019)). For the year 2017 only, both XGBoost and Lasso perform relatively well. Overall, Lasso and AutoKNN are always the top 2 winners. We speculate that data preprocessing, hyperparameter tuning, feature construction, and even the test set span can all impact the predictive performance.