THE UNIVERSITY OF CHICAGO

ENHANCING SUBSEASONAL CLIMATE FORECASTING WITH CLIMATE MODEL ENSEMBLES AND MACHINE LEARNING

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCE
IN CANDIDACY FOR THE DEGREE OF
MASTER

DEPARTMENT OF COMPUTER SCIENCE

BY
ELENA ORLOVA

CHICAGO, ILLINOIS
GRADUATION DATE 2022

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

The work in this thesis was done in collaboration with Rebecca Willett, Haokun Liu, Raphael Rossellini and Benjamin Cash. We gratefully acknowledge the support of the NSF (OAC-1934637, DMS-1930049, and DMS-2023109) and C3.ai.

ABSTRACT

Producing high-quality forecasts of key climate variables such as temperature and precipitation on sub-seasonal time scales has long been a gap in operational forecasting. Recent studies have shown promising results using machine learning (ML) models to advance sub-seasonal forecasting (SSF), but several open questions remain. First, several past approaches use the average of an ensemble of physics-based forecasts as an input feature of these models. However, ensemble forecasts contain information that can aid prediction beyond only the ensemble mean. Second, past methods have focused on average performance, whereas forecasts of extreme events are far more important for planning and mitigation purposes. Third, climate forecasts correspond to a spatially-varying collection of forecasts, and different methods account for spatial variability in the response differently. Trade-offs between different approaches may be mitigated with model stacking. This paper describes the application of a variety of ML methods to predicting monthly average precipitation and two meter temperature using physics-based predictions (ensemble members) and observational data such as relative humidity, pressure at sea level or geopotential height two weeks in advance for the whole continental U. S. Regression and tercile classification tasks using linear models, random forests, convolutional neural networks, and stacked models are considered. The proposed models outperform common baselines such as historical averages and ensemble averages. This paper further includes an investigation of feature importance and trade-offs between using the full ensemble or only the ensemble average.

CHAPTER 1

INTRODUCTION

Nowadays, weather forecasts are routinely available out to a few days [Lorenc, 1986, National Academies of Sciences, 2016, National Research Council, 2010, Simmons and Hollingsworth, 2002], and seasonal forecasts are routinely available out to a few months [Barnston et al., 2012a]. 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 climate 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 [Ocean Studies Board, National Academies of Sciences, Engineering, and Medicine, 2016]. SSF is challenging for a variety of reasons. Firstly, high-quality SSF is proven difficult to accomplish compared to both short-term weather forecasting and long-term seasonal forecasting [Vitart et al., 2012]. Due to the chaotic nature of atmosphere, weather events can not be accurately predicted beyond two weeks using dynamical models [Lorenz, 1963]. 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. However, powerful 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 hydrology, agricultural productivity, and water resource management, etc. [Klemm and McPherson, 2017]

The National Centers for Environmental Prediction (NCEP), part of the National Oceanic and Atmospheric Administration (NOAA), currently issues a "week 3-4 outlook" for the continental U.S.[1]. The NCEP outlooks are constructed using a combination of dynamical and statistical forecasts, with statistical forecasts based largely on how the local climate in the past has varied (linearly) with indices of the El Niño-Southern Oscillation (ENSO),

1. https://www.cpc.ncep.noaa.gov/products/predictions/WK34/

Madden-Julian Oscillation (MJO), and global warming (i.e., the 30-year trend).

There exists great potential to advance sub-seasonal prediction using ML techniques. A real-time forecasting competition called the Sub-Seasonal Climate Forecast Rodeo [Hwang et al., 2018], sponsored by the Bureau of Reclamation in partnership with NOAA, USGS, and the U.S. Army Corps of Engineers, illustrated that teams using ML techniques can outperform forecasts from NOAA's operational seasonal forecast system.

This paper focuses on developing ML-based forecasts that leverage ensembles of forecasts produced by NCEP in addition to observed data and other features. In contrast to past work, this paper *demonstrates that the full ensemble contains important information for subseasonal climate forecasting outside the ensemble mean.* Specifically, we consider the test case of predicting monthly temperature and precipitation two weeks in advance over 3000 locations over the continental United States using physics-based predictions, such as NCEP-CFSv2 hindcasts [Kirtman et al., 2014, Saha et al., 2014], using an ensemble of 24 distinct forecasts. We repeat this experiment for the Global Modeling and Assimilation Office from the National Aeronautics and Space Administration (NASA-GMAO) ensemble, which has 11 ensemble members [Nakada et al., 2018].

In this context, this paper makes the following contributions:

- We train a variety of ML models (including neural networks, random forests, linear regression, and model stacking) that input all ensemble member predictions as features in addition to contemporaneous observations of geopotential heights, relative humidity, precipitation, and temperature from past months to produce new forecasts with higher accuracy than the ensemble mean; forecast accuracy is measured with a variety of metrics Chapter 7. These models are considered in the context of regression and tercile classification. Systematic experiments are used to characterize the influence of individual ensemble members on predictive skill in Section 8.1.2.

- The collection of ML models employed allow us to consider different modes of account-

ing for spatial variability. ML models can account for spatial correlations among both features and responses; specifically, when predicting Chicago precipitation, our models can leverage not only information about Chicago, but also about neighboring regions. Specifically, we consider the following learning frameworks: (a) learning a predictive model for each spatial location independently; (b) learning a predictive model that inputs the spatial location as a feature and hence can be applied to any single spatial location; (c) learning a predictive model for the full spatial map of temperature or precipitation – i.e., predicting an outcome for all spatial locations simultaneously. Techniques such as positional encoding in ML models helps ensure that the right neighborhood information is used for each location, adapting to geographic features such as mountains or plains.In addition, ML models present a range of options for accounting for spatial variability, each with distinct advantages and disadvantages. Our application of model stacking allows our final learned model to exploit the advantages of each method.

- We conduct a series of experiments to help explain the learned model and which features the model uses most to make its predictions. We systematically explore the impact of using lagged observational data in addition to ensemble forecasts, positional encoding to account for spatial variations in Section 8.2.

- The ensemble of forecasts from a physics-based model (e.g., NCEP-CFSv2 or NASA-GMAO) contain information salient to precipitation and temperature forecasting besides their mean, and ML models that leverage the full ensemble generally outperform methods that rely on the ensemble mean alone in Section 8.1.1.

- Finally, we emphasize that the final validation of our approach was conducted on data from 2011 to 2020 that was not used during any of the training, model development, parameter tuning, or model selection steps. We only conducted our final assessment

of predictive skill for 2011 to 2020 after we had completed all other aspects of this manuscript. Because of this, our final empirical results accurately reflect the anticipated performance of our methods on new data.

1.1 Contributions

This thesis reflects a team effort of multiple students. Elena Orlova's specific contributions to this effort include:

1. Preparing the interpolation schemes for the ensemble members to align with a grid of the ground truth data.
2. Regression task: setting up baselines; designing U-Net and linear model experiments.
3. Setting up and implementation of the stacking model approach.
4. Feature importance experiments.
5. Designing and performing the experiments for tercile classification.
6. Full ensemble vs. ensemble average experiments, experiments with the ensemble members being sorted, top k best ensemble members experiments.
7. Temperature forecasts analysis in some regions.

CHAPTER 2

RELATED WORK

While statistical models were common for weather prediction in the early days of weather forecasting [Nebeker, 1995], purely physics-based dynamic system models have been carried out since the 1980s and have been the dominant forecasting method in climate prediction centers since the 1990s [Barnston et al., 2012b]. Skillful ML approaches are developed for short-term climate prediction [Cofino et al., 2002, Ghaderi et al., 2017, Grover et al., 2015, Herman and Schumacher, 2018, Radhika and Shashi, 2009] and longer-term weather forecasting [Badr et al., 2014, Iglesias et al., 2015, Cohen et al., 2019, Totz et al., 2017]. However, forecasting on the sub-seasonal timescale, with 2-8 week outlooks, has been considered a far more difficult task than seasonal climate forecasting due to its complex dependence on both local weather and global climate variables [Vitart et al., 2012]. Seasonal prediction also benefits from targeting a much larger averaging period.

Some ML algorithms for subseasonal forecasting use purely observational data (i.e., not using any physics-based ensemble forecasts). He et al. [2020] focuses on the analysis of different ML methods, including Gradient Boosting trees and Deep Learning (DL) for SSF. They propose a careful construction of feature representations of observational data and show that ML methods are able to outperform a climatology baseline, i.e., predictions corresponding to the T-year average at a given location and time. This conclusion is made based on the relative R^2 score that represents the relative skill against the climatology. Srinivasan et al. [2021] proposes a Bayesian regression model based on exploiting spatial smoothness in the data.

Other works use the ensemble average as features in an ML system. For example, in the subseasonal forecasting Rodeo [Hwang et al., 2018], a climate prediction challenge for the western U.S. sponsored by NOAA and the U.S. Bureau of Reclamation, simple yet thoughtful statistical models consistently outperform NOAA's dynamical systems forecasts. In partic-

ular, the authors used a stacked predictor from two nonlinear regression models and created their own dataset from climate variables such as temperature, precipitation, sea surface temperature, sea ice concentration, and a collection of physics-based forecast models including the ensemble average from various modeling centers in NMME. From the MultiLLR's analysis, the ensemble average is the first- or second-most important feature for forecasting, especially for precipitation. He et al. [2021] perform a comparison of modern ML models on the Subseasonal Experiment (SubX) project for SSF in the western contiguous United States. The experiments show that incorporating ensemble average as input features to ML models leads to a significant improvement in forecasting performance, but that work does not explore the potential value of individual ensemble members aside from the ensemble mean. Grönquist et al. [2020] notes that physics-based ensembles are computationally-demanding to produce, and proposes an ML method can input a subset of ensemble forecasts and generate an estimate of the full ensemble; then note that the output ensemble estimate has more prediction skill than the original ensemble. However, their results only cover forecasts lead up to 48 hours, which is not applicable for sub-seasonal forecasting.

This paper complements the above prior work by developing powerful learning-based approaches that incorporate both physics-based forecast models and observational data to improve SSF over the whole U.S. mainland.

CHAPTER 3

PROBLEM FORMULATION

Our goal is to predict either the monthly average precipitation or monthly average 2-meter temperature two weeks in advance (for example, we predict average monthly precipitation for February on January 15). In this section, we describe the notation used for features and responses, how spatial features are accounted for, and different formulations of the learning task.

3.1 Notation

We let T denote the number of monthly samples and L denote the number of spatial locations. In this manuscript, we consider the tasks of regression and tercile classification. We define $\delta_t = 14$ days as our forecast horizon. We define the following variables:

- $u_{t,l}^{(k)}$ is the k-th ensemble member at time t and location l, where $k = 1, \ldots, K$, $t = 1, \ldots, T$, $l = 1, \ldots, L$.
- $v_{t,l}^{(p)}$ is the p-th observational variable or other features, such as geopotential height, relative humidity, etc., at time t and location l, with $p = 1, ..., P$.
- $z_l^{(1)}, z_l^{(2)}$ represent information about longitude and latitude of location l, respectively; each is a vector. More details about this representation can be found in Section 6.
- $x_{t,l} := [u_{t,l}^{(1)}, ..., u_{t,l}^{(K)}, v_{t,l}^{(1)} ..., v_{t,l}^{(P)}, z_l^{(1)}, z_l^{(2)}]$ is a set of features at time t and location l.
- $y_{t,l}$ is the "response" – the ground truth monthly average precipitation or temperature at the target forecast time $t + \delta_t$ at location l.
- $\hat{y}_{t,l}$ is the output of a forecaster for a given task at target forecast time $t + \delta_t$ and location l.

The following variables are emphasized more later, but we introduce them to have all notations in one place:

- $s_{m,l}$ – a 30-year mean of an observed climate variable, such as precipitation or temperature at month m and location l. We also refer to this as "climatology". It is defined formally in (5.1), and for each location l, it is calculated as an average monthly precipitation or temperature over 30 years for each month.

- $\hat{s}_{m,l}$ – a 30-year mean of a *predicted* climate variable, such as precipitation or temperature at month m and location l. For each location l and each month m, it is calculated as a mean of ensemble member predictions over the training period, as defined formally in (6.6).

The target variable y is observed from 1985 to 2020. Data over time range from January 1985 till October 2005 are used for training (249 time steps), and data over November 2005 to December 2010 are used for validation and model selection (63 time steps). Data from 2011 to 2020 (or from 2011 to 2018 in case of NASA-GMAO data) are used for testing our methods after all model development, selection, and parameter tuning are completed.

In our case, the number of locations $L = 3274$, there are $K = 24$ NCEP-CFSv2 ensemble members or $K = 11$ NASA-GMAO ensemble members, and the number of features is usually $P = 41$. The details can be found in the next sections.

3.2 Models of spatial variation

We consider three different forecasting paradigms. In the first, which we call the **spatial independence** model, we ignore all spatial information and train a separate model for each spatial location. In the second, which we call the **conditional spatial independence** model, we consider samples corresponding to different locations l as independent conditioned on the spatial location as represented by features $(z_l^{(1)}, z_l^{(2)})$. In this setting, a training sample

corresponds to $(x_i, y_i) = (x_{t,l}, x_{t,l})$, where, with a small abuse of notation, we let i index a t, l pair. In this case, the number of training samples is $n = TL$. For example, consider learning a linear model. Under the spatial independence paradigm, no spatial information is utilized, and the predictor assigns different weights to different ensemble members for at each location. In contrast, under the conditional spatial independence model, the linear predictor assigns the same weights to different ensemble members regardless of the spatial location, and a weighted sum of spatial location features is added to the (fixed) weighted sum of ensemble members.

In the third paradigm, which we call the **spatial dependence** model, we consider a single training sample as corresponding to full spatial information (across all l) for a single t; that is, $(x_i, y_i) = ([x_{t,l}]_{l=1,\ldots,L}, [y_{t,l}]_{l=1,\ldots,L}])$, where now i indexes t alone. Models developed under the spatial dependence model account for the spatial variations in the features and responses. For instance, a convolutional neural network might input "heatmaps" representing the collection of physics-based model forecasts across the continental U.S. and output a forecast heatmap predicting spatial variations in temperature or precipitation instead of treating each spatial location as an independent sample.

For purposes of illustration, consider a class of predictors that are weighted sums of ensemble forecasts and how different models of spatial variation affect what those weights might be:

- under spatial independence models, the weights may vary spatially but do not account for spatial correlations in the data;
- under conditional spatial independence models, the interpretation depends on the model being trained – linear models have the same weights on ensemble predictions regardless of spatial location, while nonlinear models (e.g., random forests) have weights that may depend on the spatial location;
- under the spatial dependence models, the weights vary spatially, depend on the spatial

location, and account for spatial correlations among the ensemble forecasts and other climate variables.

3.3 Forecast tasks

The learning task can be formulated as learning a predictor $f_\theta : X \to Y$ with parameters θ. This predictor f_θ can be linear regression, the mean of ensemble members, a random forest, a convolutional neural network, or other learned models.

In particular, suppose the task is to predict the average precipitation for the next month. Then the input to the predictors can be the ensemble members $u^{(k)}$, $k = 1, \ldots, K$, climate variables $v^{(p)}$, $p = 1, \ldots, P$ such as two meter temperature, relative humidity or geopotential height, target variable lagged data (ground truth precipitation two, three, four, twelve and 24 month ago), information about location $z^{(1)}$, $z^{(2)}$, principal components (PCs) of sea surface temperature (SST) – all these features form x in our notation. Usually, the number of features is $P = 41$: four climate variables, five lagged variables, eight principal components of SST and the dimension of every vector $z^{(1)}$ and $z^{(2)}$ is twelve. Using these input features, the task is to predict the average precipitation for the next month.

We consider two different forecast tasks: regression and tercile classification.

Regression Our goal here is to predict monthly average values of precipitation and two meter temperature two weeks in the future. This is a point-forecast, providing users with the precipitation or temperature they can expect, conditional on the inputs. These models are generally trained using the squared error loss function.

Tercile classification Here our goal is to predict whether the response will be "high" (above the 66th percentile), "medium" (between the 33rd and the 66th percentile), or "low" (below the 33rd percentile). These percentiles are dependent on month and location. We compute these percentile values using 1971-2000 data from the NOAA dataset (see next

section for details), and these percentiles are computed for each calendar month m and location l pair. These models are generally trained using the cross-entropy loss function.

CHAPTER 4

DATA

Table 4.1 presents a description of variables that are used in the experiments. Historical average or climatology of precipitation and temperature is calculated using 1971-2000 NOAA data [NOAA, 2022].

Also, there are many sources of ensembles of physics-based predictions produced by forecasting systems. NMME provides forecasts from multiple global forecast models from North American modeling centers [Kirtman et al., 2014]. The NMME 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 as well as evaluate model biases. In contrast, a forecast period has real-time predictions generated from dynamic models.

Type	Variable	Description	Unit	Spatial Coverage	Time Range	Data Source
Climate variable	tmp2m	Daily average temperature at 2 meters	C°	US mainland $0.5^\circ \times 0.5^\circ$ grid	1985 to 2020	CPC Global Daily Temperature [Fan and Van den Dool, 2008]
	precip	Daily average precipitation	mm	US mainland $0.5^\circ \times 0.5^\circ$ grid	1985 to 2020	CPC Global Daily Precipitation [Xie et al., 2010]
	SST	Daily sea surface temperature	C°	Ocean only $0.25^\circ \times 0.25^\circ$ grid	1985 to 2020	Optimum Interpolation SST High Resolution (OISST) [Reynolds et al., 2007]
	rhum	Daily relative humidity near the surface	Pa	US mainland and North Pacific & Atlantic Ocean $0.5^\circ \times 0.5^\circ$ grid	1985 to 2020	Atmospheric Research Reanalysis Dataset [Kalnay et al., 1996]
	slp	Daily pressure at sea level	%			
	hgt500	Daily geopotential height at 500mb	m			
Historical	tmp2m	Daily average temperature at 2 meters	K	Globally $1^\circ \times 1^\circ$ grid	1971 to 2000	NOAA [NOAA, 2022]
	precip	Daily average precipitation	mm	Globally $1^\circ \times 1^\circ$ grid	1971 to 2000	NOAA [NOAA, 2022]

Table 4.1: Description of climate variables and their data sources. Our target climate variables for sub-seasonal climate forecast are precipitation and 2-meter temperature. Since our ground truth observations only span a short period of time, we used NOAA data to calculate for historical values from 1971 to 2000 and compute the climatology. We also perform linear interpolation on the historical values to fit the grid of the ground truth variables.

In this paper, we use ensemble forecasts from NMME models: NCEP-Climate Forecast System version 2 (CFSv2) [Kirtman et al., 2014, Saha et al., 2014], which has $K = 24$ ensemble members at a $1^\circ \times 1^\circ$ resolution over a 2-week lead time. The NCEP-CFSv2 model has two different products: we use its hindcasts from 1982 to 2010 for training, developing and tuning our models and its forecasts from March 2011 to December 2020 for final evaluation of our models.

In order to ensure our results are not unique to a single forecasting model, we also analyze output from the NASA-Global Modeling and Assimilation (GMAO) from the Goddard Earth Observing System (GEOS) model version 5 [Nakada et al., 2018], which has $K = 11$ ensemble members at a $1^\circ \times 1^\circ$ resolution over a 2-week lead time. Similarly, we use its hindcasts from 1981 to 2010 for training, developing and tuning our models and its forecasts from January 2011 to January 2018 for final evaluation.

Different ensemble members correspond to different initial conditions (initialization time) of the underlying physical model. NCEP-CFSv2 is the operational prediction model currently used by the U.S. Climate Prediction Center. The NCEP-CFSv2 forecasts are initialized every five days and include 24 ensemble members in total – four members (0000, 0600, 1200, and 1800 UTC) every fifth day, starting from the month prior to the month of forecast time. NASA-GMAO is a fully coupled atmosphere–ocean–land–sea ice model, with forecasts initialized every five days and includes 11 ensemble members (one member every fifth day).

All data are interpolated to lie on the same $1^\circ \times 1^\circ$ grid, resulting in $L = 3274$ U.S. locations. Climate variables that available daily (such as pressure at sea level or precipitation) are converted to a monthly average values. When data is available as monthly averages only, we select the month used as a predictor that only uses data available at time t and earlier, so that our forecast for time $t + \delta_t$ does not use any information about the interval $(t, t + \delta_t)$.

CHAPTER 5

PREDICTION METHODS

5.1 Baselines

Climatology (i.e., historical average) This simple baseline is the 30-year mean of a variable at a given location and month. It is the fundamental benchmark for climate predictability. In particular, for a given time t, let $m(t) := t \mod 12$ correspond to the calendar month corresponding to t; then we compute its 30-year climatology of the target variable via

$$\hat{y}^{\text{hist}}_{t,l} = s_{m(t),l}. \tag{5.1}$$

Ensemble average This is the mean of all ensemble members for each location l at each time step t:

$$\hat{y}^{\text{ens avg}}_{t,l} := \frac{1}{K}\sum_{k=1}^{K} u^{(k)}_{t,l},\ t = 1, \ldots, T,\ l = 1, \ldots, L.$$

Linear regression Finally, we consider *linear regression* model. As a baseline model, we use ensemble members as input features: $x_{t,l} = [u^{(1)}_{t,l}, \ldots, u^{(K)}_{t,l}]$. Then the model's output is

$$\hat{y}^{\text{LR}}_{t,l} := \langle \theta_l, x_{t,l} \rangle + \theta^0_l, \tag{5.2}$$

where θ_l are the trained coefficients for input features for each location l, and θ^0_l are the learned intercepts for each location l. We learn a different model for each spatial location.

5.2 Learning-based methods

Linear regression (LR) Other climate variables can be added to the input features: $x_{t,l} = [u_{t,l}^{(1)}, ..., u_{t,l}^{(K)}, v_{t,l}^{(1)}, ..., v_{t,l}^{(P)}]$. Then the model's output is

$$\hat{y}_{t,l}^{\text{LR}} := \langle \theta_l, x_{t,l} \rangle + \theta_l^0, \tag{5.3}$$

where θ_l are the trained coefficients for input features for each location l, and θ_l^0 are the learned intercepts for each location l. Because the feature vector is higher dimensional here than for the baseline, the learned θ_l is also higher dimensional here. We learn a different model for each spatial location. Since linear models are trained independently for each location, the location information does not modulate weights on ensemble members. In our experiments with linear models, we do not include positional encodings $(z_l^{(1)}, z_l^{(2)})$ as input features.

Random forest We train a random forest that uses the spatial location as additional features in addition to ensemble predictions and climate data to form the feature vector

$$x_{t,l} = [u_{t,l}^{(1)}, ..., u_{t,l}^{(K)}, v_{t,l}^{(1)}, ..., v_{t,l}^{(P)}, z_l^{(1)}, z_l^{(2)}]$$

for all location l and time t pairs. One random forest is trained to make predictions for any spatial location.

Convolutional neural network To produce a forecast map for the U.S. we adapted a U-Net architecture [Ronneberger et al., 2015], which has an encoder-decoder structure with convolutional layer blocks. The U-Net maps a stack of images to an output image; in our context, we treat each spatial map of a climate variable or forecast as an image. Thus, the

input to our U-Net is

$$x_t = [u_t^{(1)}, ..., u_t^{(K)}, v_t^{(1)}, ..., v_t^{(P)}, z^{(1)}, z^{(2)}].$$

The model output is a spatial map of the predicted response.

Nonlinear model stacking Model stacking is a popular way to improve model performance by combining the outputs of several models (usually called base models) [Pavlyshenko, 2018]. In our case, linear regression, random forests, and the U-Net are substantially different in terms of architecture and computation, and we observe that they produce qualitatively different forecasts.

We stack a linear model, random forest, and U-Net using a nonparametric approach:

$$\hat{y}_{t,l} = h(\hat{y}_{t,l}^{\text{LR}}, \hat{y}_{t,l}^{\text{RF}}, \hat{y}_{t,l}^{\text{UNET}}), \tag{5.4}$$

where h is a simple feed-forward neural network with a non-linear activation, $\hat{y}_{t,l}^{\text{LR}}, \hat{y}_{t,l}^{\text{RF}}, \hat{y}_{t,l}^{\text{UNET}}$ are the predictions of a linear model, random forest and the U-Net correspondingly. Model stacking can improve the forecast quality by combining predictions from three forecasting paradigms. One stacking model is trained to make predictions for any spatial location.

CHAPTER 6

EXPERIMENTAL SETUP

In this section, we provide details on the experimental setup including removing climatology, positional encoding, evaluation metrics and training scheme for different models. Preprocessing details are presented in Appendix C.

6.1 Positional encoding

Several of our models use the spatial location as an input feature. Rather than directly using latitudes and longitudes, we use positional encoding (PE) [Vaswani et al., 2017]:

$$z_l^{(1)}(i) = \text{PE}(l, 2i) = \sin(l/10000^{2i/d}), \tag{6.1}$$

$$z_l^{(2)}(i) = \text{PE}(l, 2i+1) = \cos(l/10000^{2i/d}), \tag{6.2}$$

where l is a longitude or latitude value, $d = 12$ is the dimensionality of the positional encoding, and i is the index $i = 1, \ldots, d$.

6.2 Training

RF is known to be yet simple but powerful ML approach, and usually it is easier to train compared to DL methods. In our experiments, we use default parameters for RF from the Scikit-learn library [Pedregosa et al., 2011]. For the U-Net, we modify an available PyTorch implementation [Yakubovskiy, 2020]. We use cross-validation (CV) and grid search to select parameters such as learning rate, weight decay (the Adam optimizer [Kingma and Ba, 2014] is used in all experiments), batch size, and number of epochs. We split train part of our dataset into 10 folds, and perform a regular 10-fold CV algorithm (without shuffling). After that, having optimal hyperparameters, we train U-Net model with the optimal parameters

on the full training dataset. The validation data is used to select features and models.

For model stacking, we apply the following procedure: the base models are first trained on half of the training data, and predicted values on the second half are used to train the stacking model. Then we retrain the base models on all the training data and apply the learned stacked model to those prediction take predictions of models that are trained on all train data, and the pretrained stacking model is applied to these predictions. This scheme is utilized since machine learning models tend to overfit on the train data (especially random forest), and the stacking model could simply learn to pay more attention to overfitted predictions. The proposed procedure helps the stacking model to generalize better. The stacking model is a simple feed-forward neural network with a non-linear activation function. The architecture details can be found in Appendix B.2.

6.3 Subtracting climatology

We evaluate our detrended predictions against the detrended true responses, defined as

$$y_{t,l}^{\text{detrended}} = y_{t,l} - s_{m(t),l}, \tag{6.3}$$

$$\hat{y}_{t,l}^{\text{detrended}} = \hat{y}_{t,l} - s_{m(t),l} \tag{6.4}$$

where $s_{m,l}$ is defined in (5.1). However, for the special case of $\hat{y}$ corresponding to the ensemble average, the ensemble members may exhibit bias, in which case we also consider

$$\hat{y}_{t,l}^{\text{detrended}} = \hat{y}_{t,l} - \hat{s}_{m(t),l} \tag{6.5}$$

where $\hat{s}_{m(t),l}$ is evaluated on the model's (ensemble average) predictions:

$$\hat{s}_{m,l} := \frac{1}{T}\sum_{t=1}^{T} \hat{y}_{t,l}\mathbb{I}_{\{m=m(t)\}},\ l = 1, \ldots, L. \tag{6.6}$$

The model climatology $\hat{s}_{m,l}$ is evaluated on the train part of data.

Note that we do not apply detrending to the input features and target variables, i.e., precipitation and temperature, when training our ML models. We subtract climatology from the model outputs only at evaluating their performances.

6.4 Evaluation metrics

Regression metrics The forecast skill of our regression models measured using the R^2 *value*. The exact formula for one location l and ground-truth $y_{t,l}$ and predictions $\hat{y}_{t,l}$ at this location is given below:

$$R_l^2 = 1 - \frac{\sum_{t=1}^{T} \left(y_{t,l}^{\text{detrended}} - \hat{y}_{t,l}^{\text{detrended}} \right)^2}{\sum_{t=1}^{T} \left(y_{t,l}^{\text{detrended}} - \bar{y}_{l}^{\text{detrended}} \right)^2}, \tag{6.7}$$

for $l = 1, \ldots, L$ and

$$\bar{y}_{l}^{\text{detrended}} = \frac{1}{T} \sum_{t=1}^{T} \hat{y}_{t,l}^{\text{detrended}}.$$

Then the average R^2 for all locations is calculated as

$$R^2 = \frac{1}{L} \sum_{l=1}^{L} R_l^2. \tag{6.8}$$

In addition to the average R^2 on the test, we also estimate the median R^2 score across all U.S. locations.

We further report the *mean squared error* (MSE) of our predictions:

$$\text{MSE}_l := \frac{1}{T} \sum_{t=1}^{T} \left(y_{t,l} - \hat{y}_{t,l} \right)^2,$$

for $l = 1, \ldots, L$, and

$$\text{MSE} = \frac{1}{L} \sum_{l=1}^{L} \text{MSE}_l. \tag{6.9}$$

We also report the standard error (SE), median, and 75th percentile of MSE_l. The standard errors provided here should be used with caution, since there are significant spatial correlations in the MSE values across locations, so we do not truly have L independent samples from an asymptotically normal distribution.

Tercile classification metrics We estimate the accuracy of our tercile classification predictions as the proportion of correctly classified samples out of all observations.

CHAPTER 7

EXPERIMENTAL RESULTS

In this section, we report the predictive skill of different models applied to SSF over the continental U.S. using NCEP-CFSv2 ensemble members for regression and tercile classification. The results for the NASA-GMAO dataset are presented in Appendix A. Recall that all methods are trained on data spanning January 1985 - October 2005, with data spanning November 2005 - December 2010 used for validation (i.e., model selection and hyperparameter tuning). Test data spanning 2011 to 2020 was **not** viewed at any point of the model development and training process, and only used to evaluate the predictive skill of our trained models on previously unseen data; we refer to this period as the "test period".

7.1 Regression

In this section, we present results for the regression task for both climate variables.

7.1.1 Precipitation regression

Precipitation regression results using NCEP-CFSv2 data are presented in Table 7.1. Our methods, especially the stacked model, outperform the baselines such as the ensemble average or historical average with statistical significance in terms of MSE. In addition, the mean and median R^2 values of the stacked model are the highest compared to all other methods. Note that the model stacking approach is applied to the models that are trained on all available features (i.e., ensemble members, positional encoding, lagged data, climate variables; linear regression is trained on all features except PE). We decide what models to include to the stacking approach based on their validation performance (for example, LR on ensemble members only performs worse than LR on all features on validation; Table 8.2). The low 90th percentile error implies that our methods not only have high skill on average, but also

that there are relatively few samples with large errors.

Model	Features	Mean R^2 (↑)	Median R^2 (↑)	Mean Sq Err (↓)	SE	Median MSE (↓)	90th prctl MSE (↓)
Baseline	Historical avg	-0.06	-0.01	2.33	0.04	1.59	4.96
	Ens avg	-0.08	0.01	2.19	0.04	1.55	4.57
	Linear Regr	-0.11	-0.07	2.26	0.04	1.54	4.72
LR	All features	-0.33	-0.25	2.71	0.05	1.91	5.45
U-Net	All features	-0.10	-0.01	2.18	0.03	1.44	4.62
RF	All features	-0.11	-0.01	2.17	0.04	1.48	4.45
Stacked	LR, U-Net, RF outputs	**0.02**	**0.04**	**2.07**	0.03	**1.42**	**4.38**

Table 7.1: **Results for precipitation regression using NCEP-CFSv2 dataset over the test period.** LR refers to linear regression on all features including ensemble members, lagged data, climate variables and SST. Model stacking is performed on models that are trained on all features. The performance of ensemble average is evaluated with the model climatology $\hat{s}$ removed.

Fig. 7.1 illustrates performance of key methods on NCEP-CFSv2 data with R^2 heatmaps over the U.S. The stacked model's heatmap reveals large areas of land where its predictive skill exceeds that of all other methods. Note that model stacking yields relatively accurate predictions even in regions where the three stacked models individually perform poorly (e.g., southwestern Arizona), highlighting the generalization abilities of our stacking approach. These plots demonstrate that random forest, U-Net and linear regression indeed produce different forecasts in terms of overall skill and spatial variation.

7.1.2 Regression of temperature

Table 7.2 shows results for 2-meter temperature regression using the NCEP-CFSv2 dataset. The learning-based models, especially random forest and stacked model, significantly outperform the baseline models in terms of MSE and R^2 score. The random forest also outperforms linear regression and the U-Net. Note that LR, U-Net and RF are trained without using SSTs information, since SSTs features yielded worse performance over the validation period (Table 8.2).

Fig. 7.2 illustrates the performance of key methods on NCEP-CFSv2 data with R^2

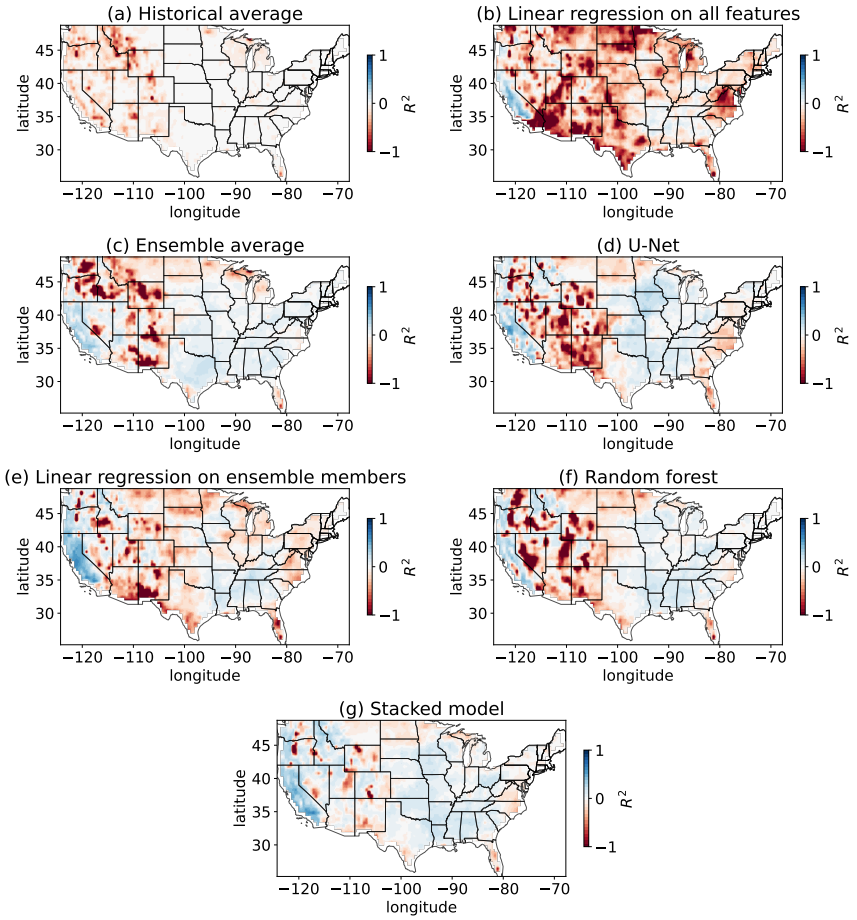

Figure 7.1: R^2 **score heatmaps of baselines and learning-based methods for precipitation regression using NCEP-CFSv2 dataset over the test period**. Positive values (blue) indicate better performance. The stacked model has the most predictive skill, including in southern Arizona, where the constituent models each perform worse.

Model	Features	Mean R^2 (↑)	Median R^2 (↑)	Mean Sq Err (↓)	SE	Median MSE (↓)	90th prctl MSE (↓)
Baseline	Historical avg	-0.66	-0.17	6.57	0.11	5.04	9.99
	Ens avg	-0.47	0.08	5.51	0.10	3.83	9.16
	Linear Regr	0.04	0.17	3.60	0.03	3.25	5.49
LR	All features wo SSTs	0.05	0.16	3.57	0.02	3.33	5.41
U-Net	All features wo SSTs	0.01	0.18	3.65	0.02	3.38	5.31
RF	All features wo SSTs	0.16	0.25	3.17	0.02	2.99	4.63
Stacked	LR, U-Net, RF outputs	**0.18**	**0.27**	**3.11**	0.02	**2.93**	**4.56**

Table 7.2: **Test results for temperature regression using NCEP-CFSv2 dataset.** LR refers to linear regression on all features including ensemble members, lagged data, land variables. Model stacking is performed on models that are learned on all features except SST. The performance of ensemble average is evaluated with a model climatology $\hat{s}$ removed.

heatmaps over the U.S. As expected, the model stacking approach shows the best results across spatial locations. We notice that there are still regions where all models tend to achieve negative R^2 score.

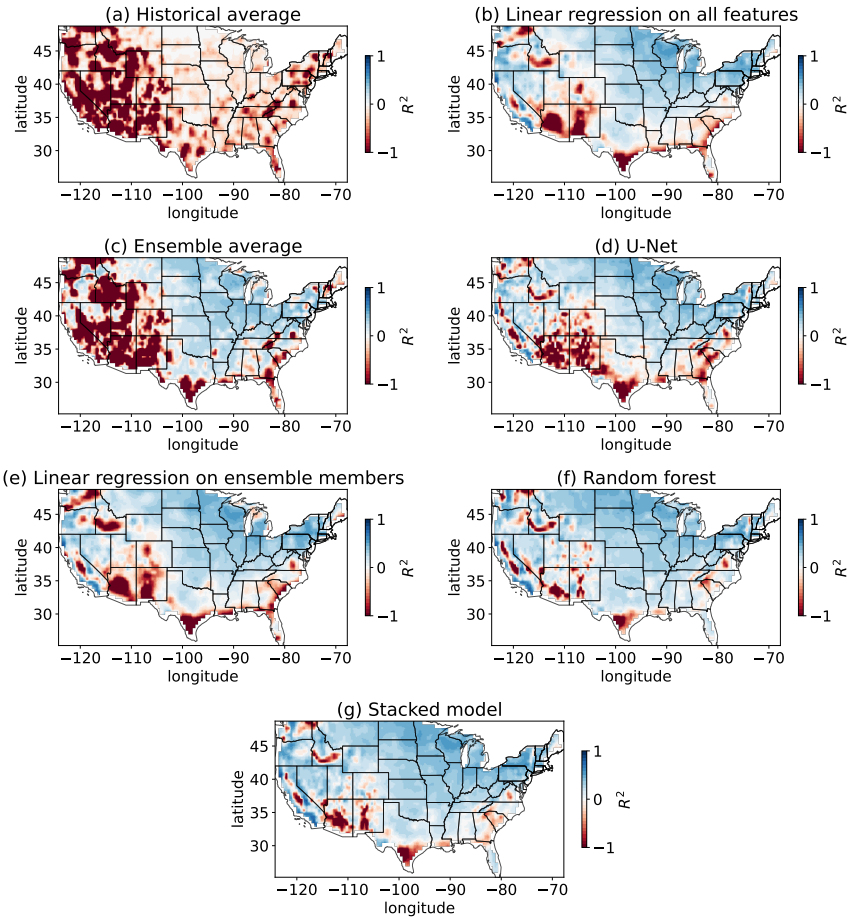

Figure 7.2: **Test R^2 score heatmaps of baselines and learning-based methods for temperature regression using NCEP-CFSv2 dataset**. Positive values (blue) indicate better performance. The stacked model's heatmap has many fewer dark red pixels than the models being stacked.

7.2 Tercile classification

In this section, we present results for the tercile classification task for both climate variables.

7.2.1 Tercile classification of precipitation

In this case, the proposed learning-based methods are trained on the classification task directly. Predictions of baselines, such as an ensemble average or a historical average, are split to three classes according to the 33rd and 66th percentile values. Note that random forest and U-Net are trained for classification using all available features. We do not notice a significant difference in performance of logistic regression on the validation if the inputs to the logistic regression are ensemble members only or ensemble members with side info. So, we use logistic regression on ensemble members only. The model stacking is applied to the logistic regression, U-Net and random forest outputs.

Table 7.3 summarizes results for NCEP-CFSv2 dataset on the test data. For this task, the learning-based methods achieve the best performance in terms of accuracy. In particular, U-Net achieves the highest accuracy score, and the performance of the stacked model is comparable with it. In general, random forest and U-Net also significantly outperform the ensemble average.

Model	Mean accuracy (↑)	Median accuracy (↑)	SE
Ens avg	38.00	37.61	0.16
Logistic Regr	41.22	40.17	0.14
U-Net	**43.88**	**42.74**	0.12
RF	42.38	41.88	0.13
Stacked	43.81	42.74	0.13

Table 7.3: **Test results for tercile classification of precipitation using NCEP-CFSv2 data.** Accuracy in % is reported. Note that for this task, our models are trained for classification directly while baselines perform regression and threshold for predicted values is applied. For stacking, logistic regression, U-Net and RF outputs are used.

The accuracy heatmaps over U. S. land are presented in the Figure 7.3 for NCEP-CFSv2 dataset. The plots corresponding to the learning-based methods show the best results, especially at the West Coast, Colorado and North America.

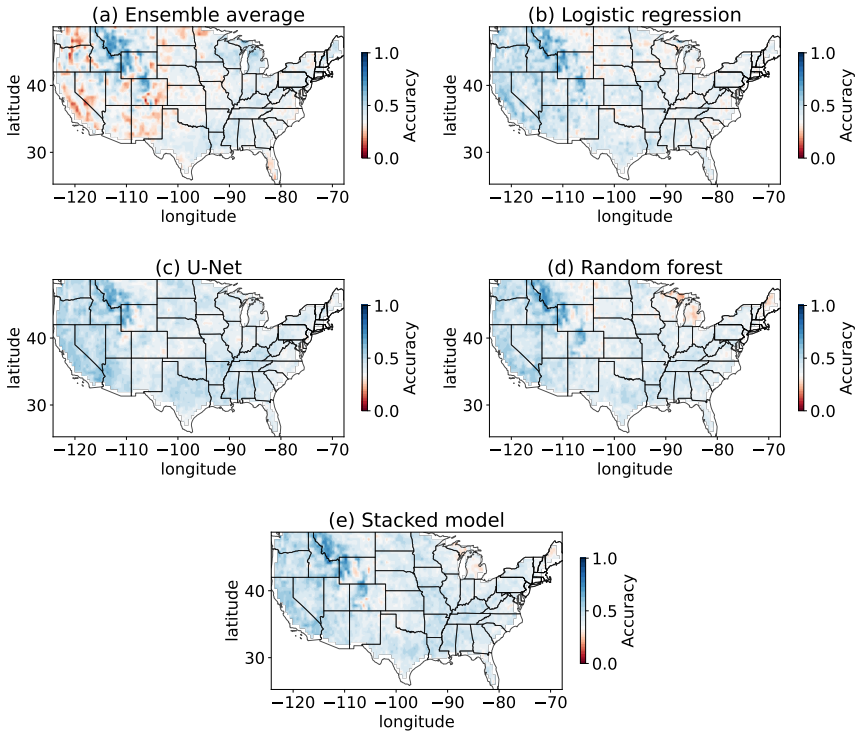

Figure 7.3: **Test accuracy heatmaps of baselines and learning-based methods for tercile classification of precipitation using NCEP-CFSv2 dataset.** The accuracy color bar is centered at $\frac{1}{3}$, what corresponds to a random guess score. Blue pixels indicate better performance, while red pixels correspond to performance that is worse than a random guess. U-Net achieves the highest accuracy score, the performance of the stacked model is lower, but the difference with U-Net is not statistically significant.

7.2.2 Tercile classification of temperature

The next task is tercile classification of 2-meter temperature. In this case, the thresholds of 33rd and 66th percentile values are applied to the regression predictions of all methods, meaning there is no direct training for a classification. Table 7.4 summarizes results for NCEP-CFSv2 data on the test. For this task, all learning-based models significantly outperform the ensemble average. The stacked model achieves the highest accuracy score.

Figure 7.4 shows accuracy heatmaps over the U.S. for different methods using NCEP-CFSv2 data. The stacked model shows the best performance across spatial locations. For example, ensemble average do not show great performance at South East and Middle Atlantic

Model	Mean accuracy (↑)	Median accuracy (↑)	SE
Ens avg	44.84	42.74	0.36
Linear Regr	57.10	54.69	0.25
LR	57.34	54.71	0.25
U-Net	53.80	50.43	0.28
RF	58.07	54.70	0.27
Stacked	**58.12**	**54.71**	0.20

Table 7.4: **Test results for tercile classification of temperature on NCEP-CFSV2 dataset.** Accuracy in % is reported. Note that for this task, our models are trained for regression and the threshold for predicted values is applied.

regions, while learning based methods demonstrate much stronger predictive skill in these areas. However, there are still some areas, such as Texas or South West region, with red pixels for all methods.

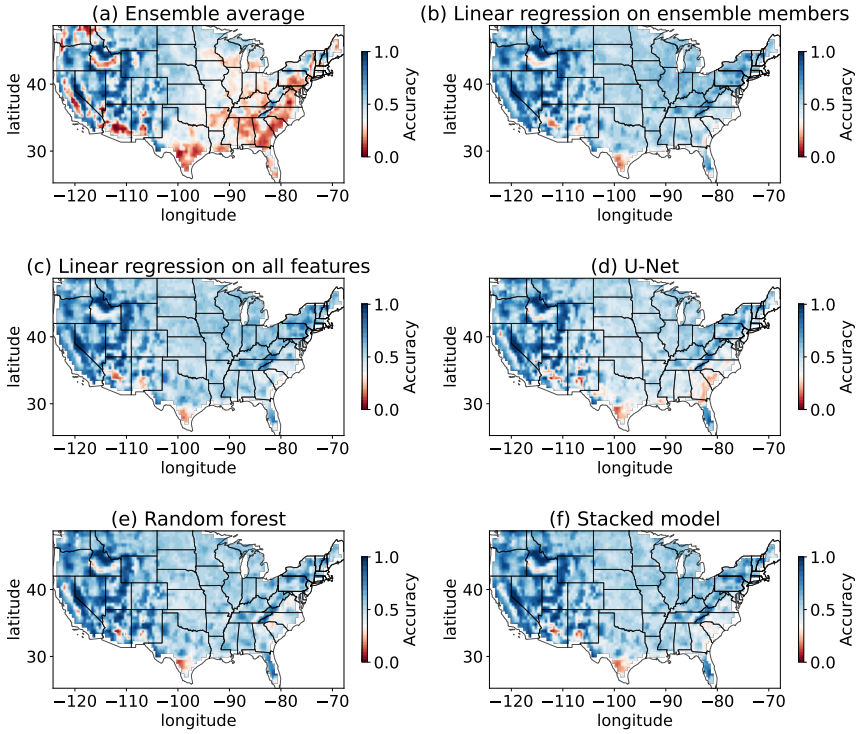

Figure 7.4: **Test accuracy heatmaps of baselines and learning-based methods for tercile classification of temperature using NCEP-CFSv2 dataset.** The accuracy color bar is centered at $\frac{1}{3}$, what corresponds to a random guess score. Blue pixels indicate better performance, while red pixels correspond to performance that is worse than a random guess. The stacked model achieves the highest accuracy score.

CHAPTER 8

DISCUSSION

8.1 The efficacy of machine learning for SSF

While climate simulations and ensemble forecasts are designed to provide useful predictions of temperature and precipitation based on carefully-developed physical models, we see that machine learning applied to those ensembles can yield significantly higher predictive skill for a range of SSF tasks. Figure 8.1 illustrates key differences between different predictive models for predicting precipitation 14 days in advance. Figure 8.1(g) shows the forecast associated with a single member of the NCEP-CFSv2 ensemble. Figure 8.1(e) illustrates the prediction of the ensemble average, and its MSE is lower compared to the individual ensemble member's error. These baselines can all be compared with the true precipitation in Figure 8.1(a). In contrast, Figure 8.1(b, d, f, h, i) show the predictions of different ML models that use all ensemble members as well as observational data: linear regression on ensemble members only (b), linear regression on ensemble members and additional features (d), a convolutional neural network (f), a random forest (h) and a stacked model (i) that gives a learned combination of the predictions of (d), (f), and (h). The learned models exhibit higher spatial frequencies and more accurately predict localized regions of high and low precipitation compared to the ensemble average. There are several hypotheses that might explain why ML is an effective approach here, and we probe those hypotheses in this section. There are several hypotheses that might explain why ML is an effective approach here, and we probe those hypotheses in this section.

8.1.1 Using full ensemble vs. ensemble average

Past works use ensemble average as an input feature to machine learning methods in addition to the climate variables [Hwang et al., 2018, He et al., 2021]. Ensembles provide valuable

information not only about expected climate behavior, but also variance or uncertainty in multiple dimensions; methods that rely solely on ensemble average lack information about this variance. The dynamical models that underpin each ensemble member may have systematic errors, either in the mean, the variability, or conditioned upon certain elements of the initial conditions, that are not readily apparent to the forecaster. While taking the average of these ensemble members may net out the deficiencies of each individual member, it is also possible that the systematic errors of each may be directly discovered and corrected by a machine learning model independently. Therefore, using a single ensemble statistic, such as the ensemble mean, as a feature may not fully capitalize on the information provided by using all the ensemble forecasts as features. In our experiments, we find that using all available ensemble members enhances the prediction quality of our approaches. As an illustration, we show results of the random forest model trained on all ensemble members and the random forest trained on the ensemble average. In addition to the full ensemble or the ensemble mean, we use other available features (as in our previous regression results). Table 8.1 demonstrates the results: using all ensemble members is crucial indeed. RF's performance on the full ensemble (and other features) is significantly better compared to RF's performance on the ensemble mean (and other features) both in terms of MSE and R^2 value. We conclude that the full ensemble contains important information for SSF aside from ensemble mean, and our models are able to capitalize on this information.

8.1.2 Learning when to trust each ensemble member

We consider the hypothesis that there is a set of k ensemble members that are always best. To test this hypothesis, we use a training period to identify which k members perform best for each location, and then during the test period, compute the average of only these k ensemble members. The performance of this approach depends on k, the number of ensemble members we allow to be designated "good", but the performance for any k never exhibited

Target	Features	Mean R^2 (↑)	Mean Sq Err (↓)	SE
Precip	Full ensemble + all features	**-0.10**	**2.17**	0.04
	Ensemble avg + all features	-0.16	2.36	0.04
	Sorted ensemble + all features	-0.17	2.30	0.04
Tmp	Full ensemble + all features wo SSTs	**0.16**	**3.17**	0.02
	Ensemble avg + all features wo SSTs	0.10	3.57	0.02
	Sorted ensemble + all features wo SSTs	0.10	3.44	0.02

Table 8.1: **Performance comparison of random forest trained using the full ensemble; using the ensemble average or using the sorted ensemble members, in addition to other features**. Scores on the test data are reported and NCEP-CFSv2 data is used.

a statistically significant improvement over the ensemble average.

If the ensemble members have different levels of accuracy over various seasons, locations, and conditions, then a machine learning model may be learning when to "trust" each member. We conduct an experiment designed to test whether it is important to keep track of which ensemble member made each prediction, or whether it is the *distribution* of predictions that is important. The modeling approach for the former would be to feed in ensemble member 1's forecast as the first feature, ensemble member 2's forecast as the second feature, etc. The modeling approach under the distributional hypothesis is to make the smallest prediction be the first feature, the second-smallest prediction be the second feature, and so on – i.e., we sort the ensemble forecasts for each location separately. Note that this entails treating the ensemble members symmetrically: the model would give the same prediction if ensemble member 1 predicted a and ensemble member 2 predicted b or if ensemble member 1 predicted b and ensemble member 2 predicted a. In statistical parlance, this is passing in the order statistics of the forecasts as the features, rather than their original ordering. (Note that for NCEP-CFSv2, ensemble forecasts are originally ordered according to the time at which their initial conditions are set [Saha et al., 2014].) We train and evaluate the random forest model with the sorted forecasts as the features and compare to our earlier results, which used the original ordering of the ensemble members. Table 8.1 illustrates results on test part. In

case of precipitation, the MSE of the sorted approach is 2.30, which is worse than the 2.17 MSE for using the original ordering. In the case of temperature forecasting, the MSE of the sorted approach is 3.44, which is much worse than the 3.17 MSE for using the original ordering. The mean R^2 of sorted approach is also lower compared to the original ordering. In both cases, the performance is better when we feed in the features in such a way that the machine learning model has an opportunity to learn aspects of each ensemble member, not merely their order statistics. Therefore, imposing a symmetric treatment of ensemble members degrades performance.

8.2 Variable importance

One aspect of the efficacy of ML for SSF is that ML models can incorporate side information (such as spatial information, lagged temperature and precipitation values, and climate variables). We explore the importance of different components of side information in this section. We see that including the observational climate variables improved the performance for both the random forest and the U-Net in case of precipitation regression task. Furthermore, including positional encodings of the locations improved the performance of the U-Net, while the principle components of the sea surface temperature didn't make much difference in case of temperature prediction. This is expected since the sea surface temperature is empirically a good predictor for precipitation at southwest but not the whole U.S.

More specifically, Table 8.2 summarizes grouped feature importance of precipitation regression using the NCEP-CFSv2 ensemble. We observe that models, in particular random forest and U-Net, trained on all available data achieve the best performance. In case of linear regression, the SSTs feature is not so helpful, but the difference is not significant. Therefore, in order to be consistent, we decide to use predictions of these models trained on all features as an input to the stacking model.

Table 8.2 summarizes grouped feature importance of temperature regression using the

Model	Features	Mean R^2 (↑)	Median R^2 (↑)	Mean Sq Err (↓)	SE	Median MSE (↓)	90th prctl MSE (↓)
LR	Ens members	-0.13	-0.08	2.11	0.03	1.53	4.63
	–"– & PE & lags	-0.11	-0.07	2.10	0.03	1.50	4.59
	–"– & land	**-0.09**	**-0.06**	**2.06**	0.03	**1.47**	**4.52**
	–"– & SSTs	-0.10	-0.07	2.08	0.03	1.47	4.61
U-Net	Ens members with PE	-0.13	-0.05	2.01	0.03	1.50	4.31
	–"– & lags	-0.08	-0.02	1.92	0.03	1.42	4.17
	–"– & land	-0.02	0.05	1.86	0.03	1.37	4.02
	–"– & SSTs	**0.00**	**0.05**	**1.83**	0.03	**1.34**	**3.94**
RF	Ens members with PE	-0.15	-0.04	2.02	0.03	1.49	4.34
	–"– & lags	-0.10	0.00	1.96	0.03	1.44	4.21
	–"– & land	-0.08	0.02	1.93	0.03	1.39	4.16
	–"– & SSTs	**-0.06**	**0.04**	**1.89**	0.03	**1.36**	**4.08**

Table 8.2: **Grouped feature importance results on validation for precipitation regression task using NCEP-CFSv2 ensemble members.** The results suggest that using additional observational information helps to improve performance of learning-based models for this task. –"– means a repetition of features that are used above. For example, in the U-Net part of the table, "–"– & PE & lags" means that ensemble members, PE, and lags are used as features and "–"– & SST" means ensemble members, PE and lags, land features and SSTs are used as features.

NCEP-CFSv2 ensemble. In this case, adding some types of side information may yield only very small improvements to predictive skill, and in some cases the additional information may decrease predictive skill. This may be a sign of overfitting. We also note that SSTs provide only marginal (if any) improvement in predictive skill, in part because Pacific SSTs are less helpful predictors away from the western U.S. It could also be that information from the SSTs is already being well-captured by the output from the dynamical models and thus including observed SSTs is not providing much in the way of additional information. Similar to the precipitation case, in order to be consistent, we decide to use predictions of these models trained on all features except SSTs as an input to the stacking model. Due to this observation, we use models trained on all side information except SSTs for evaluation on the test in case of temperature regression task.

Model	Features	Mean R^2 (↑)	Median R^2 (↑)	Mean Sq Err (↓)	SE	Median MSE (↓)	90th prctl MSE (↓)
LR	Ens members	0.35	0.40	2.19	0.02	2.00	3.47
	–"– & PE & lags	**0.37**	**0.40**	**2.12**	0.02	**1.94**	**3.30**
	–"– & land	0.36	0.39	2.14	0.04	1.94	3.40
	–"– & SSTs	0.34	0.38	2.23	0.02	1.99	3.73
U-Net	Ens members with PE	**0.33**	**0.41**	**2.22**	0.04	**2.02**	**3.47**
	–"– & lags	0.32	0.40	2.24	0.02	2.02	3.49
	–"– & land	0.31	0.41	2.26	0.02	2.08	3.48
	–"– & SSTs	0.28	0.38	2.47	0.02	2.20	3.95
RF	Ens members with PE	0.11	0.37	2.85	0.04	2.28	4.87
	–"– & lags	0.30	0.36	2.35	0.02	2.12	3.70
	–"– & land	**0.30**	**0.36**	**2.33**	0.02	**2.10**	**3.65**
	–"– & SSTs	0.28	0.34	2.42	0.02	2.17	3.83

Table 8.3: **Grouped feature importance results on validation for temperature regression task using NCEP-CFSv2 ensemble members.** The results demonstrate that using some additional information may yield only very small improvements in predictive skill, and in some cases the side information may decrease predictive skill. –"– means a repetition of features that are used above. For example, for U-Net part of the table, –"– & PE & lags means that ensemble members, PE, and lags are used as features and –"– & SSTs means ensemble members, PE and lags, land features and SSTs are used as features.

8.3 Temperature forecasting analysis

Figure 7.2 shows regions in Texas and Florida where the ensemble average and linear regression performance is poor, while a random forest achieves far superior performance. We conduct an analysis of forecasts of the ensemble average, linear regression, and random forests in these regions together with a region in Wisconsin where all methods show good performance. Figure 8.2 indicates these regions and Table 8.4 summarizes performance of different methods in these regions: the ensemble average prediction quality dramatically drops between the validation and test periods in Texas and Florida, which is not the case for the random forest.

Why does RF perform so much better than simpler methods in some regions? One possibility is that the RF is a nonlinear model capable of more complex predictions. However, if that were the only cause of the discrepancy in performance, then we would expect that the

Region location	Model	Train mean R^2 (↑)	Validation mean R^2 (↑)	Test mean R^2 (↑)
Texas	Ens avg	0.19	0.36	-1.55
	LR	0.53	0.49	-1.29
	RF	0.97	0.32	-0.33
Florida	Ens avg	0.11	0.34	-0.87
	LR	0.47	0.58	-0.56
	RF	0.97	0.36	0.11
Wisconsin	Ens avg	0.30	0.36	0.39
	LR	0.53	0.57	0.51
	RF	1.00	0.47	0.47

Table 8.4: **Train, validation, and test performance of different methods in Texas, Florida, and Wisconsin regions.** The task is temperature regression; NCEP-CFSv2 dataset is used. The performance of the ensemble average and linear regression in the test period significantly decreases in Texas and Florida, while the random forest is able to demonstrate reasonable results. All methods perform well in Wisconsin.

RF would be better not only on the test period, but on the validation period as well. Table 8.4 does not support this argument; it shows that the ensemble average and linear regression have comparable if not superior performance to the random forest during the validation period. A second hypothesis is that the distribution of the data is different during the test period during the training and validation periods. This hypothesis is plausible for two reasons: (1) climate change, and (2) the training and validation data use hindcast ensembles while the test data uses forecast ensembles. To investigate this hypothesis, in Figure 8.3 we plot the true temperature and ensemble average in the training, validation, and test periods for the three geographic regions. The discrepancy between the true temperatures and ensemble averages in the test period is generally greater than during the training and validation periods in Texas and Florida (though not in Wisconsin, a region in which validation and test performance are comparable for all methods). This lends support to the hypothesis that hindcast and forecast ensembles exhibit distribution drift, and the superior performance of the RF during the test period may be due to a greater robustness to that distribution drift.

The hindcast and forecast ensembles may have different predictive accuracies because the hindcast ensembles have been debiased to fit past climate data – a procedure not possible

for forecast data. Is the lack of debiasing of forecast data the main driver of distribution drift? To explore this, Figure 8.3 also shows the "oracle debiased ensemble average" which is computed by using the test data to estimate the forecast ensemble bias and subtracting it from the ensemble average. This procedure, *which would not be possible in practice and is used only to probe distribution drift and ensemble bias*, yields smaller discrepancies between the true data and the (oracle debiased) ensemble average during the test period. Specifically, the oracle ensemble member achieves -0.20 mean R^2 score (TX) and -0.28 mean R^2 score (FL) vs. -1.55 R^2 (TX) and -0.87 R^2 (FL) of the original forecast ensemble average. Nevertheless, even the oracle debiased ensemble average shows signs of distribution drift. Furthermore, Figure 8.3 suggests that the true distribution of temperature may be changing over time; e.g., in Texas, the true temperatures have more extreme values during the test period.

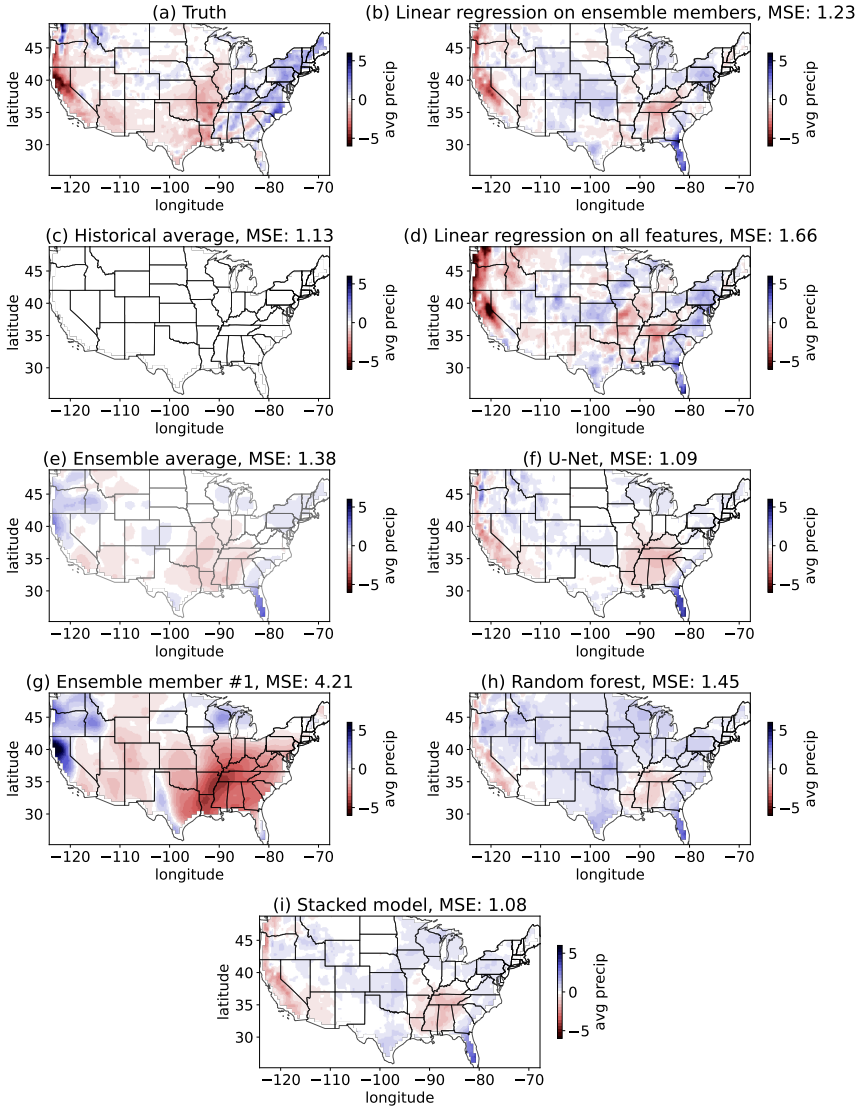

Figure 8.1: **An illustration of precipitation predictions (on the detrended scale) of different methods for February 2016 (in test period).** Individual ensemble members produce predictions with high levels of spatial smoothness, and predict extreme values in more regions. Linear regression, the random forest, the U-Net, and the stacked model all produce higher spatial frequencies. The linear regression result, which uses a different model trained for each spatial location separately, has the least spatial smoothness of all predictors; this is especially visible in the southeast and potentially does not reflect realistic spatial structure.

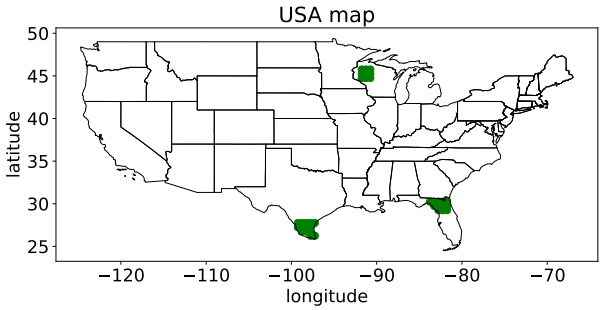

Figure 8.2: **Regions where temperature forecast is analyzed.** In Texas and Florida, the ensemble average performance drastically underperforms in the test set compared to both training and validation. In Wisconsin, all methods show good performance. See Table 8.4.

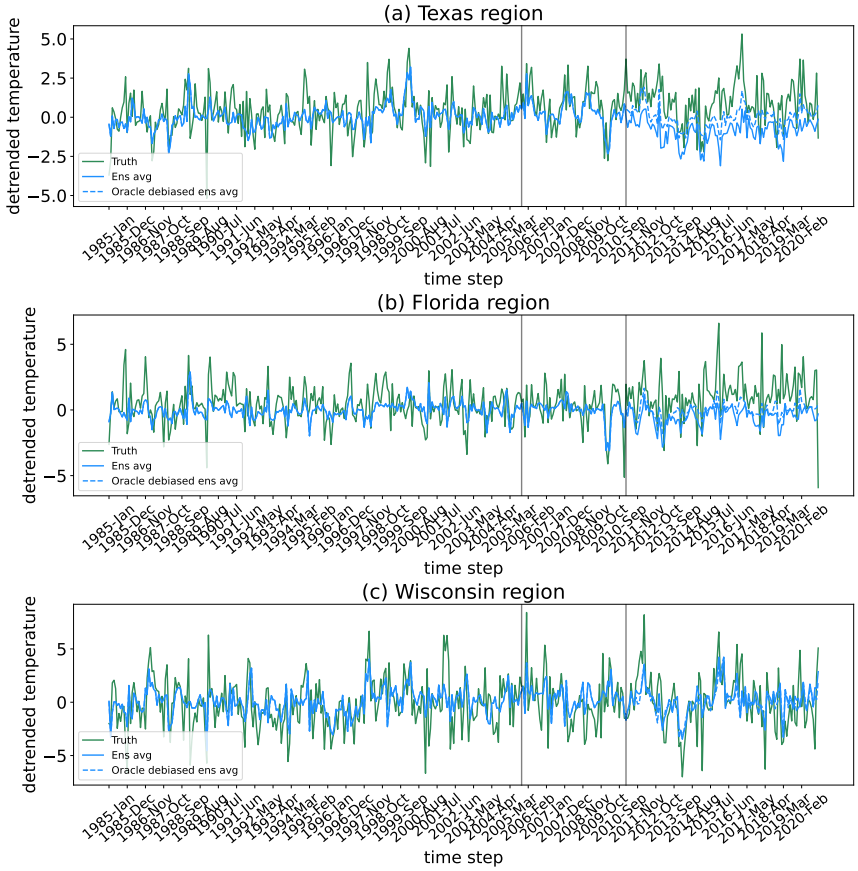

Figure 8.3: **Temperature predictions of different methods at Texas, Florida, and Wisconsin regions.** Black lines correspond to train/val and val/test splits; train and validation correspond to the hindcast regime of the ensemble, while test corresponds to the forecast regime. Figure (a) and (b) shows that the ensemble forecast is a much poorer fit to the real data than the hindcast, and there is no sign of such behavior in Figure (c). The oracle debiased ensemble average means ensemble predictions being detrended using a climatology estimated from its test predictions; and this oracle debiased ensemble average fits true response much better at Texas and Florida. It might be a sign of both distribution shift in the responses (the test period "truth" has more extreme values than in the train and validation periods) as well as a shift in the distribution of $Y|X$ (in the test period, the blue lines are a worse fit to the green lines, possibly due to the shift from hindcast to forecast ensembles).

CHAPTER 9

CONCLUSIONS AND FUTURE DIRECTIONS

This paper systematically explores the use of machine learning methods for subseasonal climate forecasting, highlighting several important factors: (1) the importance of using full ensembles of physics-based climate forecasts (as opposed to only using the mean, as in common practice); (2) the efficacy of different mechanisms, such as positional encoding and convolutional neural networks, for modeling spatial dependencies; (3) the importance of various features, such as sea surface temperature and lagged temperature and precipitation values, for predictive accuracy. Together, these results provide new insights into using ML for subseasonal climate forecasting in terms of the selection of features, models, and methods.

Our results also suggest several important directions of future research. In terms of **features**, there are many climate forecasting ensembles computed by organizations such as the NOAA and ECMWF. This paper focuses on ensembles in which ensemble members have a distinct ordering (in terms of lagged initial conditions used to generate them), but other ensembles correspond to initial conditions or parameters drawn independently from some distribution. Leveraging such ensemble forecasts, and potentially jointly leveraging ensemble members from multiple distinct ensembles, may further improve the predictive accuracy of our methods.

In terms of **models**, new neural architecture models such as transformers have shown remarkable performance on a number of image analysis tasks [Dosovitskiy et al., 2020, Carion et al., 2020, Chen et al., 2021, Khan et al., 2022] and have potential in the context of forecasting climate temperature and precipitation maps. Careful study is needed, as past image analysis work using transformers generally uses large quantities of training data, exceeding what is available in SSF contexts.

In terms of **methods**, two outstanding challenges are particularly salient to the SSF community. The first is uncertainty quantification; that is, we wish not only to forecast

temperature or precipitation, but also to predict the likelihood of certain extreme events. Second, we see in Fig. 8.3 that, at least in some geographic regions, the distribution of ensemble hindcast and forecast data may be quite different. Employing methods that are more robust to *distribution drift* [Wiles et al., 2021, Subbaswamy et al., 2021, Zhu et al., 2021] is particularly important not only to handling forecast and hindcast data, but also for accurate SSF in a changing climate.

REFERENCES

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

Anthony G. Barnston, Michael K. Tippett, Michelle L. L'Heureux, Shuhua Li, and David G. DeWitt. Skill of Real-time Seasonal ENSO Model Predictions during 2002–2011. Is Our Capability Increasing? 93:631–651, 2012a.

Anthony G Barnston, Michael K Tippett, Michelle L L'Heureux, Shuhua Li, and David G DeWitt. 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, 2012b.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12299–12310, 2021.

Antonio S Cofino, Rafael Cano, Carmen Sordo, and Jose M Gutierrez. Bayesian networks for probabilistic weather prediction. In *15th Eureopean Conference on Artificial Intelligence (ECAI)*. Citeseer, 2002.

Judah Cohen, Dim Coumou, Jessica Hwang, Lester Mackey, Paulo Orenstein, Sonja Totz, and Eli Tziperman. S2s reboot: An argument for greater inclusion of machine learning in subseasonal to seasonal forecasts. *Wiley Interdisciplinary Reviews: Climate Change*, 10 (2):e00567, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

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

Amir Ghaderi, Borhan M Sanandaji, and Faezeh Ghaderi. Deep forecast: Deep learning-based spatio-temporal forecasting. *arXiv preprint arXiv:1707.08110*, 2017.

Peter Grönquist, Chengyuan Yao, Tal Ben-Nun, Nikoli Dryden, Peter Dueben, Shigang Li, and Torsten Hoefler. Deep learning for post-processing ensemble weather forecasts. *CoRR*, abs/2005.08748, 2020. URL `https://arxiv.org/abs/2005.08748`.

Aditya Grover, Ashish Kapoor, and Eric Horvitz. 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, 2015.

Sijie He, Xinyan Li, Timothy DelSole, Pradeep Ravikumar, and Arindam Banerjee. Subseasonal climate forecasting via machine learning: Challenges, analysis, and advances. *arXiv preprint arXiv:2006.07972*, 2020.

Sijie He, Xinyan Li, Laurie Trenary, Benjamin A Cash, Timothy DelSole, and Arindam Banerjee. Learning and dynamical models for sub-seasonal climate forecasting: Comparison and collaboration. *arXiv preprint arXiv:2110.05196*, 2021.

Gregory R Herman and Russ S Schumacher. "dendrology" in numerical weather prediction: What random forests and logistic regression tell us about forecasting extreme precipitation. *Monthly Weather Review*, 146(6):1785–1812, 2018.

Jessica Hwang, Paulo Orenstein, Karl Pfeiffer, Judah Cohen, and Lester Mackey. Improving subseasonal forecasting in the western us with machine learning. *arXiv preprint arXiv:1809.07394*, 2018.

Jessica Hwang, Paulo Orenstein, Judah Cohen, Karl Pfeiffer, and Lester Mackey. 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, 2019.

Gilberto Iglesias, David C Kale, and Yan Liu. An examination of deep learning for extreme climate pattern analysis. In *The 5th International Workshop on Climate Informatics*, 2015.

Eugenia Kalnay, Masao Kanamitsu, Robert Kistler, William Collins, Dennis Deaven, Lev Gandin, Mark Iredell, Suranjana Saha, Glenn White, John Woollen, et al. The ncep/ncar 40-year reanalysis project. *Bulletin of the American meteorological Society*, 77(3):437–472, 1996.

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ben P Kirtman, Dughong Min, Johnna M Infanti, James L Kinter, Daniel A Paolino, Qin Zhang, Huug Van Den Dool, Suranjana Saha, Malaquias Pena Mendez, Emily Becker, et al. The north american multimodel ensemble: phase-1 seasonal-to-interannual prediction; phase-2 toward developing intraseasonal prediction. *Bulletin of the American Meteorological Society*, 95(4):585–601, 2014.

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

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

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

Kazumi Nakada, Robin M Kovach, Jelena Marshak, and Andrea Molod. Global modeling and assimilation office - nasa, Apr 2018. URL `https://gmao.gsfc.nasa.gov/pubs/docs/Nakada1033.pdf`.

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

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

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

NOAA. Noaa national centers for environmental information, climate at a glance: National time series. `https://www.ncdc.noaa.gov/cag/`, 2022.

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

Bohdan Pavlyshenko. Using stacking approaches for machine learning models. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*, pages 255–258. IEEE, 2018.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

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

Richard W Reynolds, Thomas M Smith, Chunying Liu, Dudley B Chelton, Kenneth S Casey, and Michael G Schlax. Daily high-resolution-blended analyses for sea surface temperature. *Journal of climate*, 20(22):5473–5496, 2007.

Yaniv Romano, Evan Patterson, and Emmanuel J. Candès. Conformalized quantile regression. In *NeurIPS*, 2019.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

Suranjana Saha, Shrinivas Moorthi, Xingren Wu, Jiande Wang, Sudhir Nadiga, Patrick Tripp, David Behringer, Yu-Tai Hou, Hui-ya Chuang, Mark Iredell, et al. The ncep climate forecast system version 2. *Journal of climate*, 27(6):2185–2208, 2014.

Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

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

Vishwak Srinivasan, Justin Khim, Arindam Banerjee, and Pradeep Ravikumar. Subseasonal climate prediction in the western us using bayesian spatial models. In *Uncertainty in artificial intelligence*, pages 961–970. PMLR, 2021.

Adarsh Subbaswamy, Roy Adams, and Suchi Saria. Evaluating model robustness and stability to dataset shift. In *International Conference on Artificial Intelligence and Statistics*, pages 2611–2619. PMLR, 2021.

Sonja Totz, Eli Tziperman, Dim Coumou, Karl Pfeiffer, and Judah Cohen. Winter precipitation forecast in the european and mediterranean regions using cluster analysis. *Geophysical Research Letters*, 44(24):12–418, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Frédéric Vitart, Andrew W Robertson, and David LT Anderson. Subseasonal to seasonal prediction project: Bridging the gap between weather and climate. *Bulletin of the World Meteorological Organization*, 61(2):23, 2012.

Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre Alvise-Rebuffi, Ira Ktena, Taylan Cemgil, et al. A fine-grained analysis on distribution shift. *arXiv preprint arXiv:2110.11328*, 2021.

P Xie, M Chen, and W Shi. Cpc global unified gauge-based analysis of daily precipitation. In *Preprints, 24th Conf. on Hydrology, Atlanta, GA, Amer. Metero. Soc*, volume 2, 2010.

Pavel Yakubovskiy. Segmentation models pytorch. `https://github.com/qubvel/segmentation_models.pytorch`, 2020.

Qi Zhu, Natalia Ponomareva, Jiawei Han, and Bryan Perozzi. Shift-robust gnns: Overcoming the limitations of localized graph training data. *Advances in Neural Information Processing Systems*, 34:27965–27977, 2021.

APPENDIX A

RESULTS FOR NASA-GMAO DATASET

A.1 Regression

A.1.1 Precipitation regression

Precipitation regression results on test are presented in Table A.1. The stacked model performance is the best in terms of MSE, however, the difference is not statistically significant compared to the baselines such as ensemble average or historical average. Model stacking is performed on top of models that are trained on all features. Other machine learning models such as RF or U-Net are not able to show the outstanding performance in terms of R^2 score too. The historical average shows the best score in terms of R^2 score, and even the ensemble average is not able to outperform it.

Model	Features	Mean R^2 (↑)	Median R^2 (↑)	Mean Sq Err (↓)	SE	Median MSE (↓)	90th prctl MSE (↓)
Baseline	Historical avg	**-0.07**	**-0.02**	2.14	0.04	**1.51**	4.40
	Ens avg	-0.11	-0.06	2.13	0.04	1.52	4.31
	Linear Regr	-0.18	-0.14	2.25	0.04	1.62	4.68
LR	All features	-0.40	-0.29	2.62	0.05	1.93	5.42
U-Net	All features	-0.19	-0.09	2.11	0.03	1.56	**4.25**
RF	All features	-0.18	-0.11	2.17	0.04	1.55	4.44
Stacked	U-Net, RF, LR outputs	-0.08	-0.06	**2.09**	0.04	1.52	4.27

Table A.1: **Test results for precipitation regression using NASA-GMAO dataset.** LR refers to linear regression on all features such as mean ensemble members, lagged data, land variables and SST. The performance of ensemble average is evaluated with a model climatology $\hat{s}$ removed.

Figure A.1 illustrates test performance of key methods on NCEP-CFSv2 data with R^2 heatmaps over the U.S. Despite the stacked model does not show the best performance in terms of mean R^2 score, its heatmap has less dark red regions, and there are areas of land where its predictive skill exceeds that of all other methods.

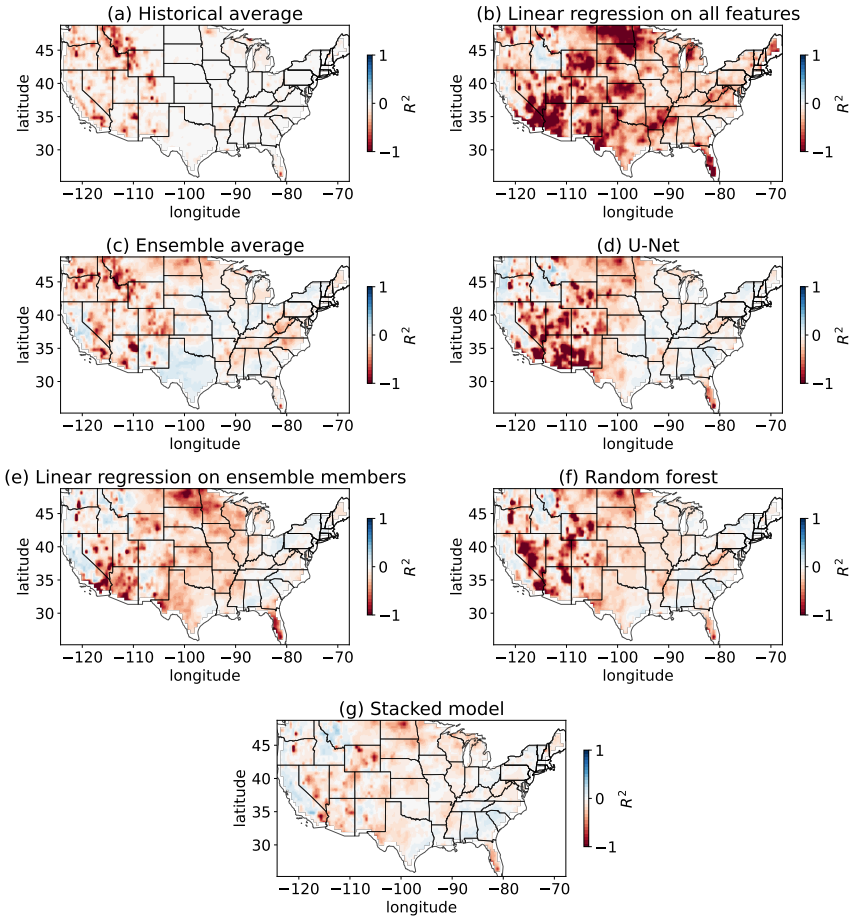

Figure A.1: **Test R^2 score heatmaps of baselines and learning-based methods for precipitation regression using NASA-GMAO dataset**. Positive values (blue) indicate better performance. The historical average achieves the best average R^2 score, however, other models have more regions with positive R^2 score.

A.1.2 Regression of temperature

Temperature regression results using NASA-GMAO ensemble members are presented in Table A.2. Random forest and U-Net outperform all baselines in terms of both R^2 score and MSE. The model stacking approach, as expected, demonstrates the best predictive skill. Note that the model stacking approach is applied to the models that are trained on all available features except SSTs (similar to NCEP-CFSv2 data).

Figure A.2 illustrates test performance of key methods on NASA-GMAO data with R^2 heatmaps over the U.S. As expected, the stacked model shows the best performance across

Model	Features	Mean R^2 (↑)	Median R^2 (↑)	Mean Sq Err (↓)	SE	Median MSE (↓)	90th prctl MSE (↓)
Baseline	Historical avg	-0.7	-0.2	6.49	0.11	5.06	9.72
	Ens avg	-0.28	0.12	4.82	0.10	3.43	7.82
	Linear Regr	0.12	0.14	3.32	0.02	3.11	4.70
LR	All features wo SSTs	0.17	0.17	3.10	0.02	3.05	4.26
U-Net	All features wo SSTs	0.06	0.12	3.40	0.02	3.27	4.52
RF	All features wo SSTs	0.20	0.22	3.03	0.02	2.94	4.25
Stacked	U-Net, RF, LR outputs	**0.21**	**0.22**	**2.94**	0.02	**2.89**	**3.97**

Table A.2: **Test results for temperature regression using NASA-GMAO dataset.** LR refers to linear regression on all features including ensemble members, lagged data, land variables and SST. Model stacking is performed on models that are learned on all features except SST. The performance of ensemble average is evaluated with a model climatology $\hat{s}$.

spatial locations. Similar to the NCEP-CFSv2 dataset, we notice that there are still regions where all models tend to achieve negative R^2 score.

A.2 Tercile classification

A.2.1 Tercile classification of precipitation

In this case, the proposed learning-based methods are trained on the classification task directly. Predictions of baselines, such as an ensemble average or a historical average, are split to three classes according to the 33rd and 66th percentile values. Note that random forest and U-Net are trained for classification using all available features. We do not notice a significant difference in performance of logistic regression on the validation if the inputs to the logistic regression are ensemble members only or ensemble members with side info. So, we use logistic regression on the ensemble members only. The model stacking is applied to the logistic regression, U-Net and random forest outputs.

Table A.3 summarizes results for NASA-GMAO dataset on the test data. For this task, the stacking models achieve the best performance in terms of accuracy. The random forest, the U-Net model and logistic regression outperform the ensemble average.

The accuracy heatmaps over U. S. land are presented in the Figure A.3 for NASA-GMAO

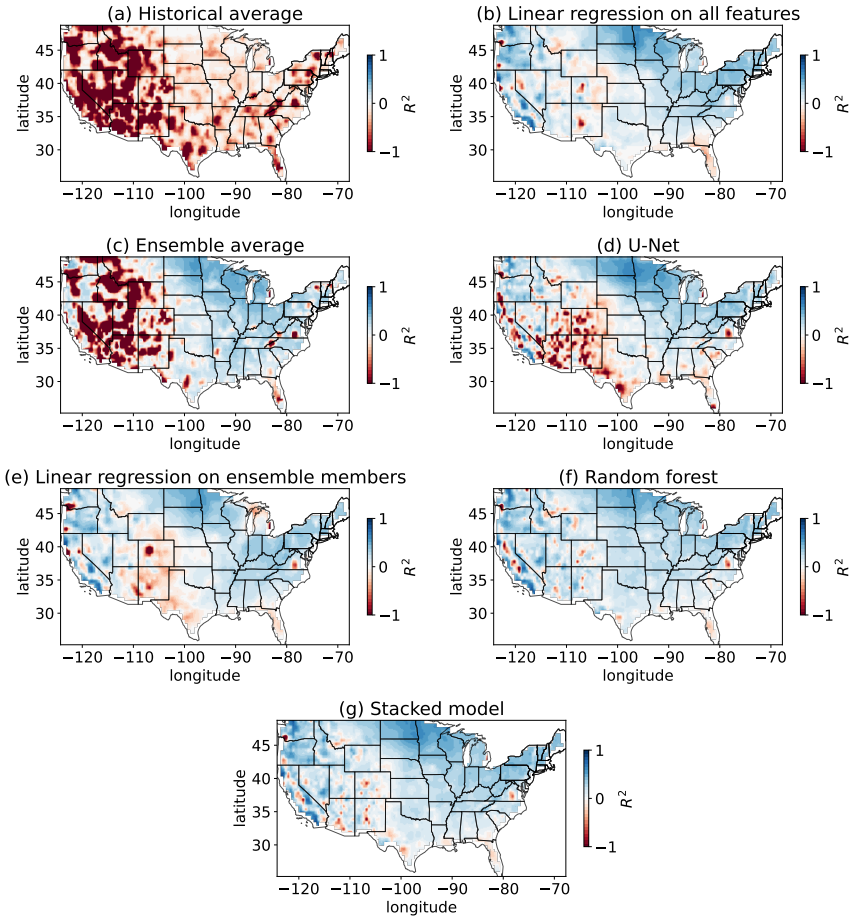

Figure A.2: **Test R^2 score heatmaps of baselines and learning-based methods for temperature regression using NASA-GMAO dataset**. Positive values (blue) indicate better performance. The stacked model's heatmap has many fewer dark red pixels than the models being stacked.

Model	Mean accuracy (↑)	Median accuracy (↑)	SE
Ens avg	38.64	37.65	0.14
Logistic regr	41.51	40.00	0.16
U-Net	40.53	40.00	0.11
RF	40.79	40.00	0.14
Stacked	**42.08**	**41.18**	0.14

Table A.3: **Test results for tercile classification of precipitation on NASA-GMAO data.** Accuracy in % is reported. Note that for this task, our models are trained for classification directly while baselines perform regression and threshold for predicted values is applied. For stacking, logistic regression, U-Net and RF outputs are used.

dataset. The plots corresponding to the learning-based methods show the best results, the ensemble average's figure has the most of dark regions.

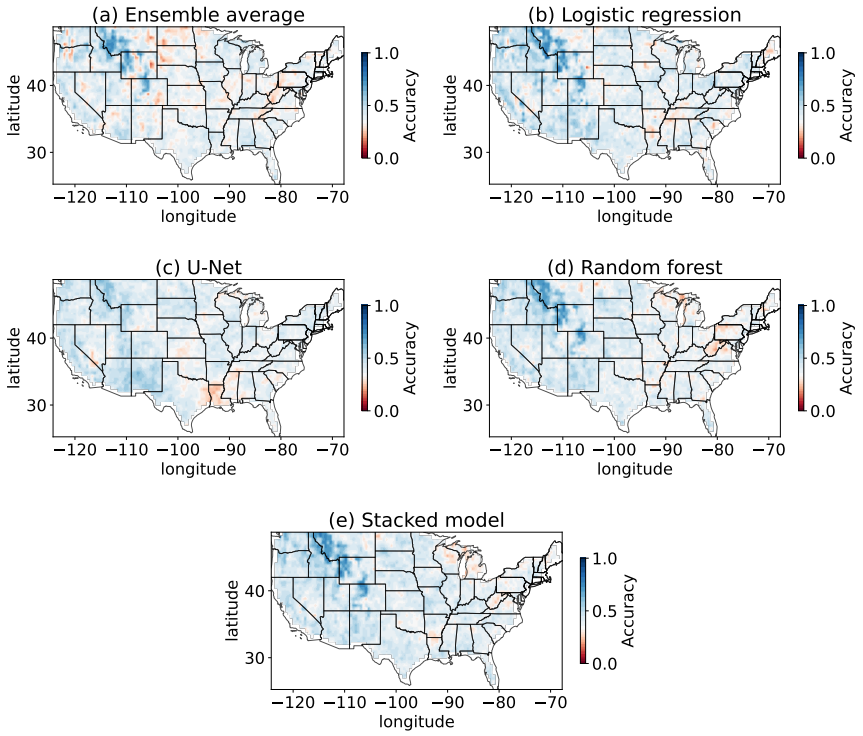

Figure A.3: **Test accuracy heatmaps of baselines and learning-based methods for tercile classification of precipitation using NASA-GMAO dataset.** BThe accuracy color bar is centered at $\frac{1}{3}$, what corresponds to a random guess score. Blue pixels indicate better performance, while red pixels correspond to performance that is worse than a random guess. The stacked model achieves the best accuracy score.

A.2.2 Tercile classification of temperature

The next task is tercile classification of 2-meter temperature. In this case, the threshold is applied to the regression predictions of all methods, meaning there is no direct training for a classification. Table A.4 summarizes results for NASA-GMAO dataset on the test data. For this task, the learning-based methods achieve the best performance in terms of accuracy, especially linear regression using NASA-GMAO ensemble members and all additional features (except SSTs). In general, all learning-based models significantly outperform the ensemble

average.

Model	Mean accuracy (↑)	Median accuracy (↑)	SE
Ens avg	52.23	49.41	0.25
Linear Regr	57.75	54.11	0.25
LR	**58.97**	**55.29**	0.25
U-Net	55.64	51.76	0.27
RF	58.78	55.29	0.26
Stacked	58.72	54.12	0.25

Table A.4: **Test results for tercile classification of temperature on NASA-GMAO data.** Accuracy in % is reported. Note that for this task, our models are trained for regression and the threshold for predicted values is applied.

Figure A.4 shows accuracy heatmaps over the U.S. for different methods using NASA-GMAO data. In this case, linear regression achieves the best scores. Other learning-based methods outperform the ensemble average too, especially at the West.

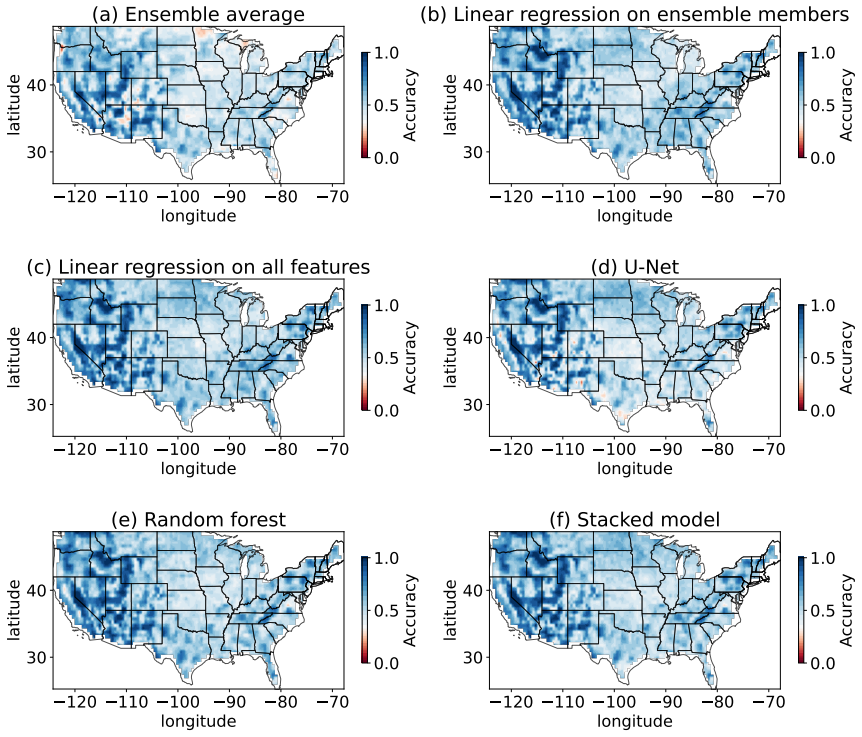

Figure A.4: **Test accuracy heatmaps of baselines and learning-based methods for tercile classification of temperature using NASA-GMAO dataset.** The accuracy color bar is centered at $\frac{1}{3}$, what corresponds to a random guess score. Blue pixels indicate better performance, while red pixels correspond to performance that is worse than a random guess. LR shows the best accuracy, and other learning-based models demonstrate comparable performance.

APPENDIX B

ARCHITECTURE DETAILS

B.1 U-Net details

The U-Net has residual connections from layers in the encoder part to the decoder part in a paired way so that it forms a U-shape. Figure B.1 shows the architecture of the U-Net. The U-Net is a powerful deep convolutional network that is widely used in image processing tasks such as image segmentation and image classification.

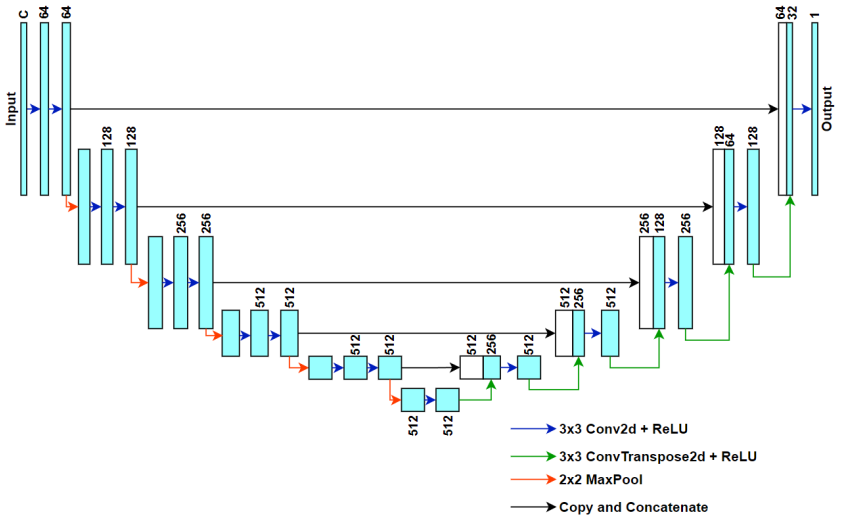

Figure B.1: U-Net architecture with input channels = C. C is the number of input channels, which in our case equals the number of ensemble members plus all climate data.

Our U-Net differs from the original U-Net by modifying the first 2D convolutional layer after input. Since our input channels can be different when we choose different subset of features or different ensemble (NCEP-CFSv2 or NASA-GMAO), this 2D convolutional layer is used to transform our input with C channels into a latent representation with 64 channels. The number of channels C depends on which ensemble we are using, and what task we are performing. For example, for precipitation tasks using the NCEP-CFSv2 ensemble, the

input channels include 24 ensemble members, 5 lagged observations, 4 other observational variables, 8 principal components of SST, and 24 positional encoding, resulting in 65 channels in total. For temperature tasks using the NASA-GMAO ensemble, there are only 11 ensemble members, and we don't include SSTs information, hence there's only 44 channels in total. The other following layers are using the same configurations with the standard U-Net Ronneberger et al. [2015].

We also perform careful hyperparameter tuning for the U-NET. In particular, we run 10-fold cross validation on our training set, and use grid search for tuning learning rate, batch size, number of epochs, and weight decay. Since we use different loss functions for different forecast tasks and different number of input channels for NCEP-CFSv2 and GMAO-GMAO ensemble, we run hyperparameter tuning with the same cross validation scheme separately for these tasks. For example, for precipitation regression, we choose from 100, 120, 150, 170, 200, 250 epochs; batch size may be equal to 8, 16, 32; learning rate values are chosen from 0.0001, 0.001, 0.01; weight decay can be 0, 1e-3, 1e-4. For instance, in case of NCEP-CFSv2 precipitation regression, the optimal parameters are 170 epochs, batch size 16, learning rate 0.0001, and weight decay 1e-4. For temperature regression using the same data, we use the following parameters: 100 epochs, batch size 16, learning rate 0.001, and weight decay 1e-3. For tercile classification of precipitation, the best parameters are 80 epochs (we chose from 60, 70, 80, 90, 100 epochs during classification), batch size 8, learning rate 0.001, and no weight decay.

B.2 Stacking model details

The stacking model is a simple one layer neural network with 100 hidden neurons and sigmoid activation function for regression and softmax for classification. We use an implementation from Scikit-learn library [Pedregosa et al., 2011]. We choose 100 neurons based on the stacking model performance on the validation data (we also tried 50, 75, 100 and 120 neurons).

The stacking model demonstrates stable performance in general, but for 100 neurons it usually achieves the best results. We use "lbfgs" optimizer from quasi-Newton methods for the regression tasks, and SGD optimizer for classification tasks.

APPENDIX C

ADDITIONAL PREPROCESSING DETAILS

Random forest and U-Net require different input formats. For U-Net, all input variables have natural image representation except SSTs and information about location. For example, ensemble predictions can be represented as a tensor of shape (K, W, H), where K corresponds to the number of ensemble members (or number of channels of an image), W and H are width and height of the corresponding image. In our case, $W = 64$ and $H = 128$.

Sea surface temperatures There are more than 100 000 SSTs location available. We extract the top eight principal components. Principal component analysis fits on train part and then is applied for the rest of the data. In case of U-Net, we deal with PCs of SSTs by adding additional input channels that are constant across space, each channel corresponds to one of PCs. Random forest can use PCs from SSTs directly with no special preprocessing.

Normalization We apply channel-wise min-max normalization to the input features at each location based on training part of the dataset in case of U-Net. As for normalization of the true values, min-max normalization is applied for precipitation and standardization is applied for temperature. This choice affects the final layer of U-Net model too: for the precipitation regression task, the sigmoid activation is used, and no activation function is applied for temperature regression. For the staking model, we apply min-max normalization to both input and target values.