THESIS

USING MACHINE LEARNING TO IMPROVE VERTICAL PROFILES OF TEMPERATURE AND MOISTURE FOR SEVERE WEATHER NOWCASTING

Submitted by

Jason D. Stock

Department of Computer Science

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Summer 2021

Master's Committee:

Advisor: Charles Anderson
Co-Advisor: Imme Ebert-Uphoff

Shrideep Pallickara
Christian Kummerow

ABSTRACT

USING MACHINE LEARNING TO IMPROVE VERTICAL PROFILES OF TEMPERATURE AND MOISTURE FOR SEVERE WEATHER NOWCASTING

Vertical profiles of temperature and moisture as provided by radiosondes are of paramount importance to forecasting convective activity, yet the National Weather Service radiosonde network is spatially coarse and suffers from temporal paucity. Supplementary information generated by numerical weather prediction (NWP) models is invaluable—analysis and forecast profiles are available at a high sampling frequency and horizontal resolution. However, numerical models contain inherent errors and inaccuracies, and many of these errors occur near the surface and influence the short-term prediction of high impact events such as severe thunderstorms. For example, the convective available potential energy and the convective inhibition are highly dependent on the near-surface values of temperature and moisture. To address these errors and to create the most useful vertical profiles of temperature and moisture for severe weather nowcasting, we explore a machine learning approach to combine satellite and surface observations with an initial NWP profile.

In particular, we explore deep learning to improve vertical profiles from an NWP model, which is the first known work to do so. Using initial profile predictions from the Rapid Refresh (RAP) model, corresponding surface products from the Real-Time Mesoscale Analysis (RTMA), and satellite data from the Geostationary Operational Environmental Satellite (GOES)-16 Advanced Baseline Imager, we train variations of fully-connected and convolutional neural networks with custom knowledge guided loss functions to produce enhanced profiles. We evaluate the success of our approach by comparing estimates with ground truth radiosonde observations (RAOB)s and their derived indices for samples collected between January 1, 2017 and August 31, 2020. The proposed Residual U-Net architecture shows a 26.15% reduction in error over the profiles relative to

the RAP errors, with the greatest improvements in the mid- to upper-level moisture. Furthermore, we detail the importance of the GOES-16 channels and assess our model under different meteorological conditions, finding: 1) no bias of seasonality; 2) training with additional samples, even in cloudy conditions, to be beneficial; and 3) sounding locations with more samples and higher initial errors to have greater improvement. As such, this work is targeted to aid forecasters concerned with severe convection make more precise predictions, thereby enhancing the nation's readiness, responsiveness, and resilience to high-impact weather events.

ACKNOWLEDGEMENTS

I would first like to acknowledge and thank my advisors, Dr. Chuck Anderson and Dr. Imme Ebert-Uphoff, for their calm guidance, supportive feedback, and encouragement to maintain a healthy life balance during these pressing times of a pandemic. My immense gratitude goes to Dr. Jack Dostalek, the Principal Investigator, for conceptualizing the research proposal, and Dr. Louie Grasso who both provided valuable domain expertise, helpful resources, and continuous feedback. Thank you to Dr. Grasso and John Dandy for assisting with data acquisition and accelerating our research efforts. I also greatly appreciate the Cooperative Institute for Research in the Atmosphere for setting up and sharing their new GPU system, which truly simplified and helped to run my experiments. Lastly, I thank my friends and family for their extensive support and understandings as I work toward completing my degree. This project was supported under NOAA Grant NA19OAR4320073.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Environmental instability of the atmosphere of the Earth often leads to the initiation of thunderstorms, which are often the source of severe weather events around the world. The most severe storms, such as those accompanied by tornadoes, downbursts, and extreme hail events, impose threats to lives and property. For meteorologists to forecast these events and conduct accurate near-term convective threat assessments, it is of paramount importance to have accurate boundary-layer thermodynamic and kinematic profiles. Radiosondes are the traditional gold-standard for accurate profiles, as they come from observational measurements from weather balloons as they ascend in the atmosphere. However, the National Weather Service (NWS) radiosonde network is spatially and temporally sparse with routine launches only twice a day, around 0000 and 1200 UTC from 92 stations across the United States.

The spatiotemporal sparsity of radiosondes imposes a significant challenge to accurately depict the change in environmental conditions over a select window in space and time. For example, a study of severe thunderstorms between 1999 and 2009 across the northeastern United States concludes that the temporal discrepancy and $250\,\mathrm{km}$ distance between sounding sites is a leading cause for introducing errors in the representation of near-storm environments [1]. Another study from the Severe Environmental Storms and Mesoscale Experiment indicates that radiosonde observations separated by 3 hours with distances of a few hundred kilometers would have been needed to resolve the changes in temperature, moisture, and wind distributions that occurred prior to the Wichita Falls tornadoes [2, 3]. Unfortunately, the manual labor and environmental impact associated with radiosonde launches limit the ability to increase the number of launches.

As a result, it is common for forecasters to use thermodynamic profiles generated from Numerical Weather Prediction (NWP) models as they are available at a high spatial and temporal resolution. These models are borne from vast amounts of data, including radiosondes, with sophisticated physics and are often validated against real-world observations; however, they include

errors resulting from uncertain initial conditions, necessary assumptions, and from the mathematics of prognosis. For operational meteorologists, it is critical to have accurate profiles especially during lower-predictability, high-impact weather events, where variance in the simulated environmental conditions can yield substantial differences among expected hazards. In particular, severe weather regimes characterized by limited buoyancy and strong vertical wind shear have presented great predictability challenges–owing to the highly sensitive nature of the vertical thermodynamic structure in the boundary layer. The accuracy at the surface through the boundary-layer (which influence vertical buoyancy distributions) is of particular interest with direct impact on a range of applications, including: severe convection [4], fire weather [5], aviation [6], agriculture [7], wind energy [8], and many more. A great deal of these minor sensitivities are within the scale of expected error, which highlights the critical importance of improving the thermodynamic boundary-layer structure within NWP simulations.

To address the errors inherent to NWP simulations and to create more useful vertical profiles of temperature and moisture, we explore various deep learning models to combine near-surface observations and satellite retrievals to improve initial NWP profiles. Figure 1.1 illustrates a high level overview of our approach with the flow of data and primary data sources. Using profile output from the Rapid Refresh (RAP) NWP model with corresponding surface observations from the Real-Time Mesoscale Analysis (RTMA) and satellite imagery from the Geostationary Operational Environmental Satellite (GOES)-16 Advanced Baseline Imager (ABI), we train variations of fully-connected and convolutional neural network architectures. These networks learn to make predictions on collocated data samples by minimizing the difference between the radiosonde observations (RAOB)s and profile estimates during training. At run time, a given sample from a location within the regime can be applied to the network to produce improved estimates of temperature and moisture. Our method is quantitatively evaluated under various meteorological conditions, and is shown to generally improve the accuracy of vertical profiles.

The rest of this thesis is organized as follows. Chapter 2 provides background to NWP models, including their initialization and why they have inherent errors. Secondly, the background

of artificial neural networks and their application in atmospheric science are discussed. Chapter 3 describes the primary datasets and preprocessing techniques that we use. Chapter 4 lays out the specifics of the various neural network architectures and how the data is used for training. Chapter 5 discusses the experimental procedure, network and training configurations, and relevant metrics to evaluate the different network architectures. Chapter 6 outlines the results of each architecture and details the best model setup with examples of profile estimates. In Chapter 7 we evaluate this model under different meteorological conditions and with different data features to better understand the model's estimates. Chapter 8 shows the results of deriving profile indices from machine learning estimates and then assess the performance of directly estimating the indices. Lastly, Chapter 9 summarizes the findings in this thesis, including the limitations and technical challenges, and then suggests potential avenues of future research.

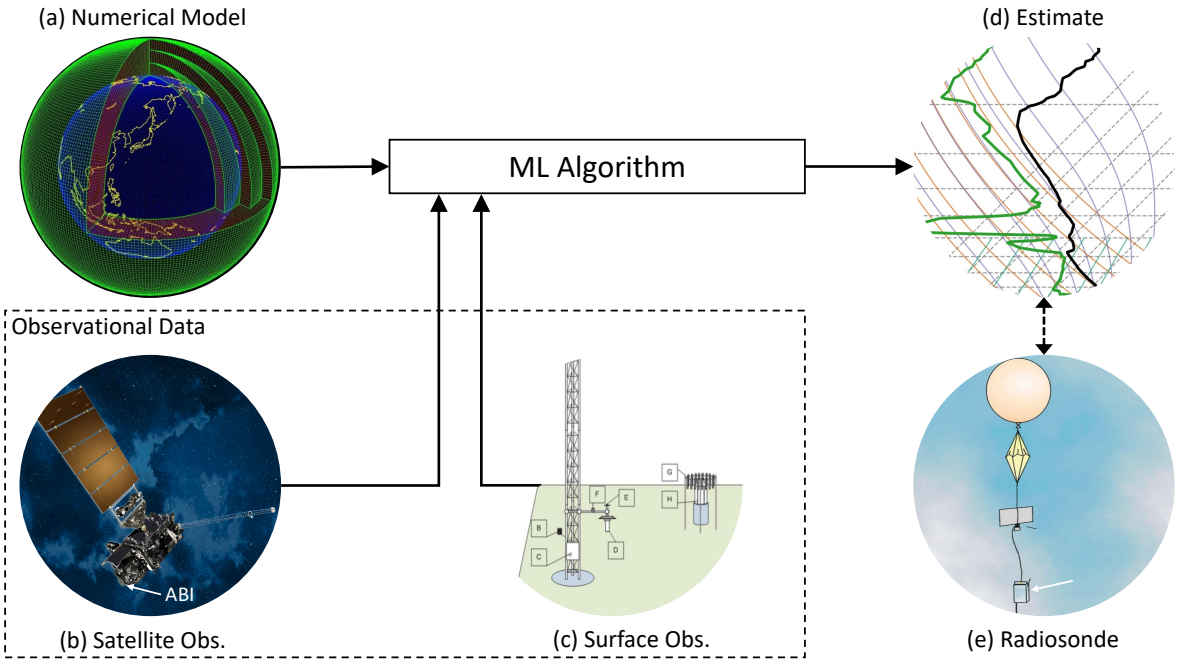


Figure 1.1: High level flow diagram and primary data sources for the proposed approach to improving vertical profiles. Initial profiles of temperature and moisture from (a) numerical weather prediction model along with collocated observations of (b) satellite imagery and (c) near-surface measurements are input to a machine learning algorithm. The output from the algorithm is a (d) profile estimate that is more similar to ground truth (e) radiosonde observations.

Chapter 2

Background

2.1 Numerical Weather Prediction

Numerical Weather Prediction (NWP) is a method to forecast the weather and atmospheric conditions using mathematical equations based on the laws of physics. These methods assimilate initial observational and boundary conditions and use systems of governing equations detailing fluid motion, thermodynamics, and radiative processes to predict the weather. There exist a number of forecast models from the National Oceanic and Atmospheric Administration (NOAA), including: the Global Forecast System (GFS), Weather Research and Forecasting (WRF) model, Rapid Refresh (RAP), and many others. While many of these models are operationally sufficient, they also have inherent sensitivities that influence their accuracy. The equations used to simulate the environments are not very precise as they can not be solved directly, and small errors propagate over time due to the lack of precision for fractional numbers in computers. Additionally, the full extent in the initial state is never entirely known since initial observations have marginal variability and are never complete. As a result, forecasters will use ensembles of these NWP models to produce a more reliable forecast; although, the accuracy of each model underpins the overall accuracy.

In this thesis, we seek to improve the profiles produced by an individual model, namely the RAP, as to support the National Centers for Environmental Prediction and the National Weather Service in their mission to provide accurate forecasts and watches for severe weather events over the United States. A detailed description of this model and our use of the data is outlined in Section 3.2. Improvements to NWP typically occurs in successive development of new models and data assimilation techniques [9], resulting in a the large variety of models and versions. With respect to vertical profiles specifically, there is limited research that explores improving the explicit output of an NWP model, especially using machine learning techniques.

The most relevant work for this thesis comes from Schmit *et al.* [10] and their validation of the legacy atmospheric profile (LAP) algorithm using GOES-16. In that work, temperature and dewpoint temperature profiles are derived from clear-sky radiances from GOES-16 ABI and initial guess profiles from the GFS 6- to 12-hour forecasts. In the retrieval, temperature and dewpoint are used in a linear regression model that is solved with general least squares, and the regressed profiles are then used as an initial guess in a 1-dimensional variational physical retrieval. Validation of temperature and dewpoint profile errors are calculated by comparing the differences with radiosonde observations (RAOB)s. Not only does this retrieval technique improve the first-guess GFS profiles, especially the middle and upper troposphere moisture, but it replaces the derived products generated using the sounder on the GOES series satellites operating before GOES-16. We employ a similar approach to validate our algorithm using RAOBs as ground truth observations. In this work, however, we explore machine learning techniques, specifically artificial neural networks, to improve retrievals of profiles from the RAP model.

2.2 Artificial Neural Networks

Artificial neural networks are a class of machine learning algorithms with a structure that is loosely based on the understanding of the mammalian's biological nervous system. Research surrounding the artificial neural network has a long history. In 1943 the first computational model of a neuron was proposed by McCulloch-Pitts [11], which paved the way for biological processes and various feed-forward neural networks. Detailed mathematical models followed with works from individuals such as Hebb [12] on theories of neuron excitements and the connections between neurons as well as the first supervised learning strategy, Rosenblatt [13] with work outlining the concept of the perceptron for supervised learning of binary classifiers, Rumelhard and Hinton [14] who popularized the backpropagation algorithm for practical training of multi-layer networks, and many others.

Through advancements in theory and computational hardware engineering, modern neural networks have become rapidly popular with applications in machine vision, medical applications, fi-

nancial applications, agricultural applications, and more. In atmospheric science specifically, there are pioneering works in using neural networks for retrieving vertical profiles of temperature and moisture from radiometric measurements [15–17] and satellite observations [18, 19], estimating atmospheric conditions from RAOBs [20–22] and numerical weather prediction models [23–26]. However, to the best of our knowledge this is the first work that uses ABI with neural networks together with an NWP first guess fields to improve the representation of atmospheric humidity. Therefore, we leverage the strengths apparent in the following approaches as means to experiment and build upon.

Improvements to retrieval techniques used by ground-based radiometers are made using standard fully-connected neural networks. As with the present study, these works compare results with RAOBs using error distributions and correlation analysis. Chakraborty *et al.* [15] uses radiometer derived brightness temperatures at various frequencies and other surface meteorological sensors to produce profiles of temperature and moisture at a number of vertical heights. Results show the neural network to outperform radiometric quadratic regression and piece-wise linear regression. Similarly, Yan *et al.* [16] demonstrate the effectiveness of radiometric atmospheric profiling with various regularization techniques and Knupp *et al.* [17] specify the use of neural networks as the standard approach on retrieving profiles from the ground-based microwave radiometer profiler for operational activities.

The use of NWP output is employed with neural networks for various applications. Lima *et al.* [23] train a fully-connected network using atmospheric variables from the WRF model with observational ground data to forecast surface solar irradiance. Håkansson *et al.* [24] use pressure and temperature variables at different vertical levels from the European Centre for Medium-Range Weather Forecast's 91-level short-range forecast along with other atmospheric variables to train neural networks for cloud top height retrievals. Veillette *et al.* [25] explore convolutional neural networks for creating synthetic radar precipitation mosaics by incorporating lightning information, visible and infrared satellite imagery from GOES-13, and fields sampled from the RAP numerical model as input to the model. Data from the RAP are also used by Lagerquist *et al.* [26], where

1-dimensional profiles of atmospheric state variables and water species with other environmental variables are input to a convolutional network to emulate and accelerate a shortwave radiative-transfer model.

To capture spatial patterns for correction in the profiles, we also consider the use of convolutional neural networks. These architectures are typically feed-forward networks with alternating convolutional and subsampling layers; however, their implementation and objective usability differs from the standard fully-connected networks. 1-dimensional convolutions are often employed for signal processing–a domain similar to that of modeling vertical profiles of temperature and moisture. The first proposed application of which was on Electrocardiogram signals for classification [27], and their experimental results yield superior performance over other classification methods. We focus on signal-to-signal regression as opposed to classification, but the usability of convolutions remains consistent.

There are many varieties of convolutional networks, including the U-Net architecture [28] that we adapt for vertical profiles. [26] provides a reference to this architecture using profiles from the RAP as input. Another variant is the Residual U-Net that is used for audio super-resolution of 1-dimensional signals. Kuleshov *et al.* [29] use a standard U-Net with an additional additive connection of the input to increase the sampling rate of signals such as music or speech. Their architecture predicts missing samples of linearly interpolated low-resolution signals to match those at high quality. The authors find that the learning-based algorithm outperforms general purpose interpolation schemes due to their ability to capture the domain specific appearance of natural signals. With respect to thermodynamic profiles, the RAOBs are often of higher resolution than estimates. As such, an improvement to a profile can be thought of as interpolating the missing values in the simulated environment, and we consider a similar architecture.

For this thesis, we explore the class of networks used in the domain of signal processing in the following categories: linear regression, fully-connected, and variations of convolutional networks. The primary reasons for this is two-fold, that is (a) the 1-dimensional thermodynamic profiles represent signals measured by radiosondes or approximated by NWP models, and (b) neural networks

for signal-to-signal processing are often designed analogously to the problem of mapping NWP profiles to ground truth RAOBs. The aforementioned works motivate how some of the datasets described in Chapter 3 are used with neural networks, and we build upon these techniques by adapting their structures for our application (detail in Chapter 4).

Chapter 3

Dataset Details

The data collected are between January 1, 2017 and August 31, 2020 over the Continental United States (CONUS). These are consolidated to focus primarily on the central region spanning between North Dakota and Texas, totaling 18 sounding locations. We focus on this region for two primary reasons, namely (a) the region, known as tornado alley, attributes the largest number of severe weather events in the United States; and (b) including additional locations, e.g. coastal or mountainous areas, are not helpful given our total data size of $38,373$ samples as the thermodynamics can be significantly different. Individual data samples are collocated using the supplementary launch details of the RAOBs. Specifically, information from each data source are extracted following the observation nearest the release time and spatial region of a given radiosonde.

3.1 Radiosonde Observations

A *radiosonde* is a small telemetry instrument sent airborne under a weather balloon that is filled with helium gas to collect data relating to different levels in the atmosphere. As the radiosonde ascends it measures vertical distributions of pressure, temperature, and relative humidity, while altitude and winds are derived from GPS location information. The radiosonde transmits recordings to a ground station via radio signals every second. The National Weather Service routinely launches radiosondes twice-daily from a network of 92 stations across the United States, albeit we only consider a subset of these locations surrounding the central states. These launches are coordinated to simultaneously occur shortly before 0000 and 1200 UTC, and together provide a general representation of the state of the atmosphere on that day.

A radiosonde observation (RAOB) when displayed on a Skew-T Log-P thermodynamic diagram is often useful for meteorologists to make short-term predictions of the weather. Figure 3.1 is an example diagram with the temperature, T, and dewpoint, T_d, profiles from April 2, 2017 23:03 UTC over Aberdeen, South Dakota (ABR). There are 5 fixed components that comprise

the diagram, namely: temperature, pressure, dry adiabats, moist adiabats, and mixing ratio. The temperature lines (gray, dashed) are drawn at a linear 45° angle with an increase in value from the upper-left to lower-right. Horizontal pressure lines (gray, dashed) are drawn on a logarithm scale to follow the decrease in atmospheric pressure with the increase in altitude. Dry adiabats, drawn in orange, increase in value from lower-left to upper right and represent the rate at which an unsaturated parcel of air cools as it rises in the atmosphere. The moist adiabats are drawn in blue and follow the lapse rate at which a saturated parcel of air changes as it ascends vertically. Lastly, the mixing ratio is shown in light green from 1000 to $600\,\mathrm{mb}$, and denotes the amount of water vapor in the environment at the point where the dewpoint temperature intersects this line.

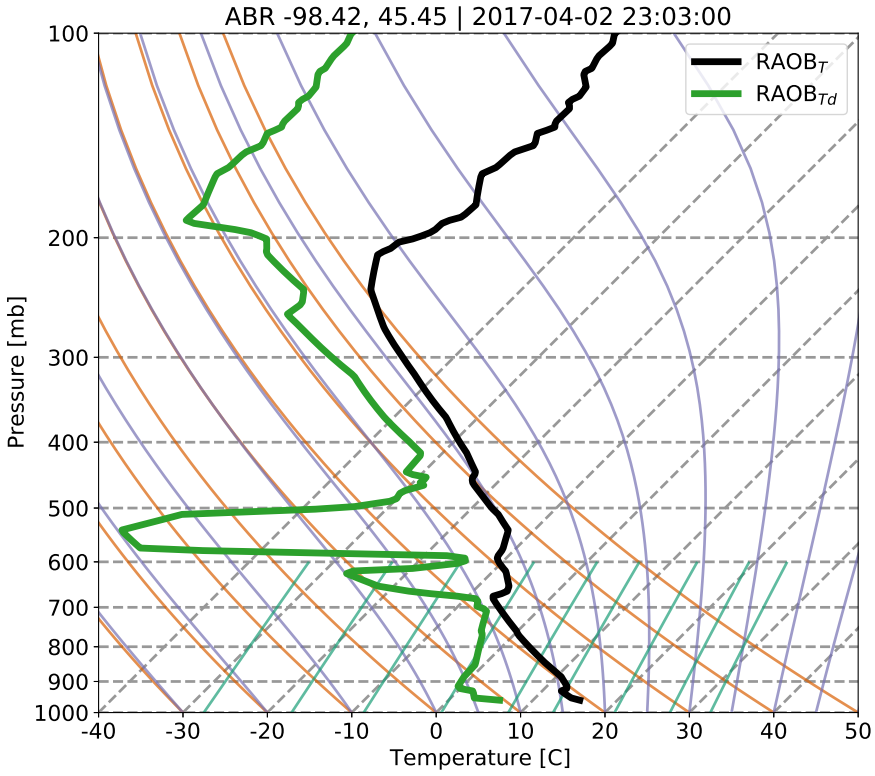


Figure 3.1: A standard Skew-T Log-P diagram of temperature, T, and dewpoint temperature, T_d, profiles for a radiosonde observation over Aberdeen, South Dakota (ABR).

Meteorologists use the plotted RAOB to obtain a wealth of information concerning upper-air conditions. The diagram can be used to assess the stability of the atmosphere, cap strength, con-

vective temperatures, and much more. Additionally, various derived indices of atmospheric conditions, such as Convective Inhibition (CIN), Convective Available Potential Energy (CAPE), or Total Precipitable Water (TPW), can be computed by using the observed temperature and dewpoint values. Therefore, the importance of accuracy in the profile is essential for accurate forecasts.

We gather data from the NOAA ESRL/GSD radiosonde archive for the 18 locations of interest, yielding a total of $38,373$ samples. As observed in Figure 3.2, not every site shares an equal number of launches as additional observations are made when atmospheric conditions are of interest, whereas other samples may have been missed or removed for miscellaneous reasons. Every observation within the database undergoes an extensive quality assurance analysis to check for and correct various hydostatic consistencies prior to data acquisition [30]. This procedure is designed to detect erroneous data and inconsistencies between observed values and reported heights. For example, correcting heights and temperature for two consecutive large deltas, checking for superadiabatic lapse rates, and general sanity checks. Corrections are commonly made when enough data is present; however, error checks that fail will either set measurements to missing or be removed all together.

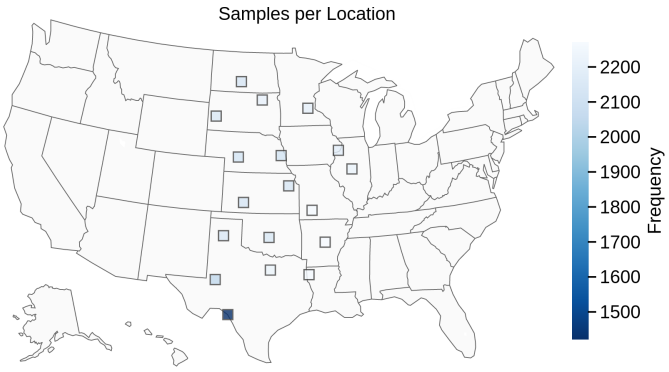


Figure 3.2: Map representing the 18 different locations within the Central United States and how the $38,373$ total samples are distributed. Not every location contains the same number of data samples.

The acquired data is minimally processed after quality assurance with conversions of variables, the removal of missing values, and formatting of data to follow a consistent vertical spacing. Foremost, the moisture profile is converted from dewpoint depression to dewpoint temperature

in degrees Celsius. The conversion is done to provide a more interpretable field that can later be visualized on a Skew-T diagram or used to compute products from the profile (*e.g.*, CIN or CAPE). Thereafter, we discard samples with measurements that contain missing values, and lastly, transform the data to linear intervals with respect to geopotential height. The primary reason for transforming data is to obtain profiles of a fixed dimension that are used with a neural network. The difference between profiles, be at separate locations or differing times, are influenced by diverse landscapes, weather systems, and larger global patterns. However, the reported geopotential height along the profile remains consistent at each location. We leverage this observation to interpolate the mandatory and significant levels of each atmospheric variable for every profile.

To assess an appropriate top layer boundary we first find the daily composite mean of the $100\,\mathrm{mb}$ surface for geopotential height from the National Centers for Environmental Prediction (NCEP) Reanalysis[1] using data for the month of July over the years between 2010 to 2019. A pressure level of $100\,\mathrm{mb}$ is chosen as it is the convention to display soundings up $100\,\mathrm{mb}$ on standard Skew-T diagrams. Furthermore, we use data from July as it is the month with the highest $100\,\mathrm{mb}$ heights and provides a better upper level bound. Over the entire CONUS domain, the $100\,\mathrm{mb}$ height is between 16–17 km with a variation of $240\,\mathrm{m}$, which is about 1.5% of the mean height at the surface. This is expected, as the atmosphere acts as a low-pass filter for vertically propagating waves. Therefore, we empirically determine to use equally spaced layers from the surface to the top, which we define as $17\,\mathrm{km}$ above the surface. Each atmospheric variable is then linearly interpolated to 256 levels with layers separated by $66.7\,\mathrm{m}$.

3.2 The Rapid Refresh

The Rapid Refresh (RAP) is an hourly-updated assimilation and modeling system with the capability of providing NWP guidance out to 18 hours for short-term forecasts and situational-awareness analyses over North America [31]. The RAP uses the community-driven Advanced

[1]The data are from the NCAR - reanalysis project. NCEP Reanalysis data provided by the NOAA/OAR/ESRL PSL, Boulder, Colorado, USA, from their web site at https://psl.noaa.gov/

Research version of the WRF model [32] for numerical weather prediction, as well as the NOAA Gridpoint Statistical Interpolation analysis system [33–35] for data assimilation and to initialize the model. Additional components ranging from observational satellite radiances from GOES and radar reflectivity assimilation via latent heating to meteorological aerodrome reports for cloud and precipitation hydrometeor assimilation are included during initialization. The RAP uses a $13\,\mathrm{km}$ horizontal grid spacing with a hybrid sigma vertical coordinate consisting of 51 levels on a Lambert Conformal Conic map projection. The sigma coordinate defines levels by a ratio of pressure at a given point in the atmosphere to the pressure of the surface directly below, thus simplifying the lower boundary conditions by following the topographical variances of the Earth's surface.

Using the release time and longitude/latitude of a given RAOB we locate the nearest RAP file and extract total pressure, temperature, specific humidity, and geopotential height at every level over the launch location. Moisture content is represented by specific humidity, but to better compare the profiles with the RAOBs, we make the conversion to dewpoint temperature. The first step is to convert the specific humidity, q, to vapor pressure, e:

$$e = \frac{pq}{\epsilon + (1-\epsilon)q}, \tag{3.1}$$

where $\epsilon = 0.622$ is the ratio of gas constants for dry air and water vapor and p is the total pressure (dry air plus water vapor). To calculate a given dewpoint temperature, t_d, in terms of vapor pressure, an expression for the dependence on e and t_d is needed. An accurate and well recognized empirical approximation relating the two is the Magnus formula represented in terms of saturation vapor pressure, e_s, and temperature, t:

$$e_s = C\exp\Big(\frac{At}{B+t}\Big). \tag{3.2}$$

A parcel of air becomes saturated at temperature t_d when an air parcel at temperature t and pressure p is cooled isobarically. The relation to vapor pressure as expressed by $e = e_s(t_d)$ is

substituted into (3.2) to compute t_d. Now solving accordingly:

$$t_d = \frac{B\ ln(\frac{e}{C})}{A - ln(\frac{e}{C})}. \tag{3.3}$$

Alduchov and Eskridge [36] recommend when working with the standard surface and upper-air data to use an approximation with the following coefficients: $A = 17.625$, $B = 243.04°$C, and $C = 610.94$ Pa. Note that C has units of Pa and q, the specific humidity in e is unitless (kg/kg). Therefore, we plug the RAP values into (3.1) to get the vapor pressure, which is also in Pa, then use (3.3) to retrieve the dewpoint temperature in degrees Celsius.

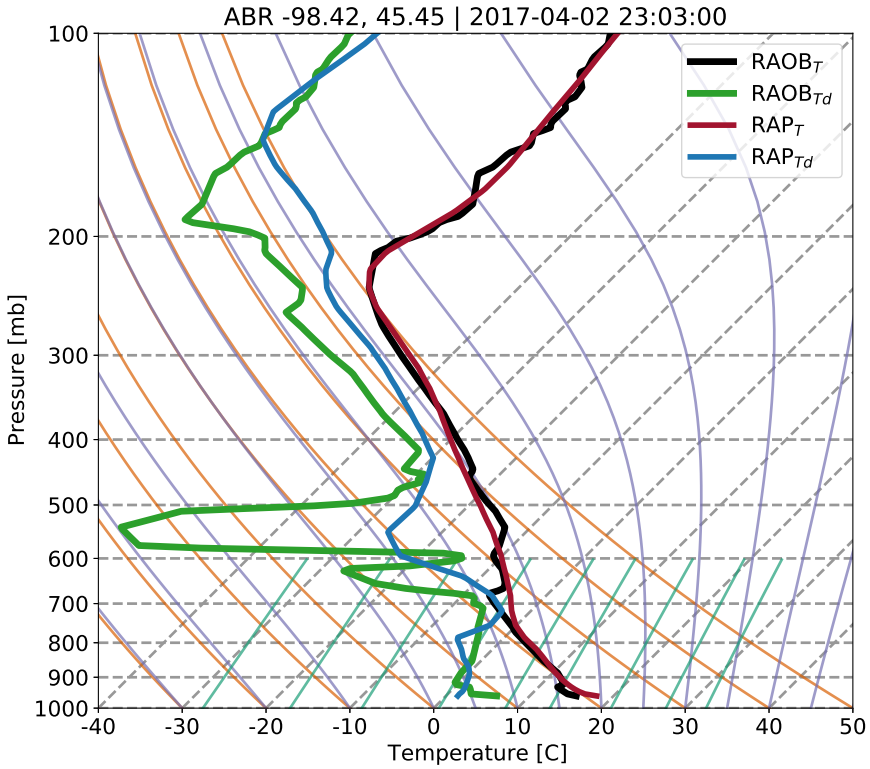


Figure 3.3: A standard Skew-T Log-P diagram of temperature, T, and dewpoint temperature, T_d, profiles for a radiosonde observation (RAOB) and collocated profiles from the Rapid Refresh (RAP) NWP model over Aberdeen, South Dakota (ABR).

The four 1-dimensional profile components (total pressure, temperature, dewpoint temperature, and geopotential height) are linearly interpolated, similar to the RAOBs, to contain 256 levels with

a top boundary layer of $17\,\mathrm{km}$ above the surface. As a result, the RAP profiles align closely to the RAOBs with corresponding measurements at every level. Figure 3.3 illustrates the RAP profiles of T and T_d overlaid on the RAOB shown in Figure 3.1. The difference is most significant in dewpoint temperature as the RAP is a smooth approximation for the high variability of the profile. Differences in the temperature profile are evident, yet less significant. However, the critical region where errors exists are at the surface and capping inversion. An *inversion* implies that temperature increases with an increase of altitude (contrary to the normal temperature decrease), and a *capping inversion* is an inversion that caps a convective planetary boundary layer and limits the vertical development of clouds. In the given example, we see the RAP misrepresents these fields with a higher surface temperature and weak inversion around $700\,\mathrm{mb}$ relative to the RAOB.

3.2.1 Profile Comparisons of the RAP and RAOB

An initial analysis shows a difference between the profiles exists with greater magnitudes in error at the surface and in the mid- to upper-level moisture. Error is measured by calculating the root-mean-squared error (RMSE) in two different ways. The first is over every vertical level individually:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(t_{ij} - y_{ij})^2} \;\; | \; 1 \leq j \leq 256, \tag{3.4}$$

where n is the total number samples, y_{ij} is the predicted value, and t_{ij} is the actual target value for the i^{th} sample at vertical level j. The second way is by taking the RMSE over all k vertical levels at once:

$$RMSE = \sqrt{\frac{1}{n}\frac{1}{k}\sum_{i=1}^{n}\sum_{j=1}^{k}(t_{ij} - y_{ij})^2}. \tag{3.5}$$

In the context of comparing the RAP and RAOB profiles to compute a baseline, we use y_i and t_i, respectively. The baseline errors in the profiles over all samples in the dataset can be seen in Figure 3.4. This provides a view of the overall error and locations in the vertical where the errors are high/low. The solid lines represent the RMSE found using (3.4), whereas the dashed lines and values in the legend are those found using (3.5).

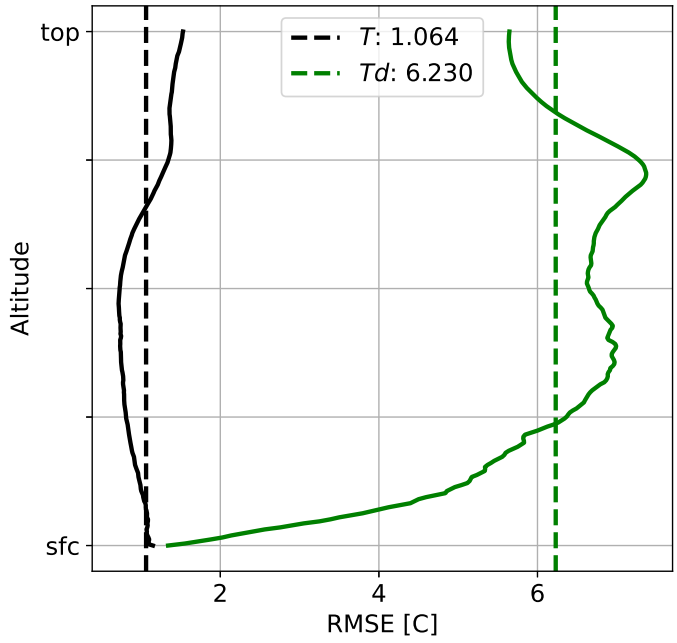


Figure 3.4: Root-Mean-Squared Error calculated for the temperature, T, and dewpoint temperature, T_d, profiles averaged over all vertical levels (dashed lines) and separately at each level for all samples in the dataset (solid lines). Each sample is interpolated between the surface (sfc) and $17\,\mathrm{km}$ above the surface (top).

Evidently, the error in the temperature profile is significantly lower than in the dewpoint profile at every vertical level. The temperature profile has a small increase in error at the surface, then decreases with altitude until about $8.5\,\mathrm{km}$ above the surface, where the error begins to increase with height. Conversely, the dewpoint temperature profile errors increases with height from the surface until the mid- and upper-level, where the errors remain constant or decrease with height. However, the error in the baseline for dewpoint is nearly six times greater at every level besides the surface and near-surface measurements.

3.3 The Real-Time Mesoscale Analysis

The Real-Time Mesoscale Analysis (RTMA) is a high-spatial and temporal analysis system run hourly to produce analysis of near-surface weather conditions [37]. Its primary component is the NCEP Gridpoint Statistical Interpolation (GSI) [38] package used in the incremental two-

dimensional variational mode. The GSI equations are solved on a Lambert Conformal grid to assimilate observational data. Hourly analyses are performed using measurements from near-surface synoptic observations, aviation routine weather reports, the Mesoscale Network (Mesonet), ships, buoys, and Coastal Marine Automated Network stations.

The system uses observations captured ± 12 minutes centered around the analysis time to produce a CONUS grid output with a grid spacing of $2.5\,\mathrm{km}$. As a result, analyses of $2\,\mathrm{m}$ temperature, $2\,\mathrm{m}$ specific humidity, $2\,\mathrm{m}$ dewpoint temperature, $10\,\mathrm{m}$ wind components, and surface pressure are assembled with corresponding estimates of uncertainly for each variable. This study considers temperature, dewpoint temperature, and surface pressure as variables of interest as it relates to measurements captured by radiosondes. Several layers of quality control are preformed during RTMA analysis to filter out erroneous data, including removing preflagged observations, verifying threshold constrained error checks, removing static and dynamic blacklisted observations, and using only trusted providers and stations. This control procedure reduces the need for further preprocessing of the data.

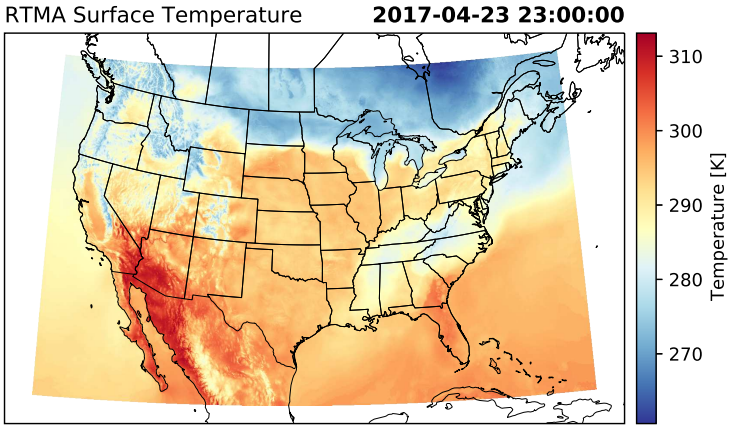


Figure 3.5: RTMA surface temperature over the CONUS domain.

Near-surface temperature, dewpoint temperature, and pressure from the RTMA are of particular interest as they relate to the variables from the RAOBs. Figure 3.5 illustrates the temperature in Kelvin over the entire CONUS grid. The majority of this grid is not used, and we consider only

the RTMA at times and locations of individual radiosonde launches. Using the release time from a given RAOB, we identify data files for the three variables with the closest analysis time. RTMA samples from past or future observations are used in situations where radiosondes are launched irregularly or beyond the top of the hour. Although rare, data samples that do not have an analysis within an hour time frame are discarded. The temporally aligned samples are then cropped to contain only the region surrounding the launch site. Specifically, we extract a 3×3 patch of size $56.25\,\mathrm{km}^2$ from each of the RTMA variables with the center point closest to the latitude and longitude of the RAOB. While this data is spatial by design, we collapse dimensionality and take the mean of each variable separately, *i.e.* of the 3×3 array, as we are more concerned with the vertical resolution. Furthermore, the horizontal area need not be large as the surface observations are nearest to the radiosonde launch location.

3.3.1 Surface Comparisons of RTMA and Vertical Profiles

To understand how the RTMA values compare to the surface values from the RAOB and RAP we contrast the distribution of the difference between samples. Using the center point from the RTMA patches and surface values from each profile we compute the difference individually for all samples with:

$$D^{RAOB} = \phi^{RTMA} - \phi^{RAOB} \tag{3.6}$$

$$D^{RAP} = \phi^{RTMA} - \phi^{RAP}, \tag{3.7}$$

where ϕ denotes the column vectors of meteorological variables (pressure, temperature, and dew-point temperature), meaning the first element, D_1, represents the difference at the surface for the first collocated sample of RTMA, RAOB, and RAP, D_2 represents the second sample, and so on for n samples in the dataset.

Figure 3.6 shows the scatter plot of D^{RAP} and D^{RAOB}, where each point represents a given sample for one of the three meteorological variables. Units along the axes are shared, native to the variable of interest. The sample difference is plotted against each other such that when they are

identical, values will follow a one-to-one line. Additionally, if there is little difference between the surface observation of the vertical profiles and the RTMA, then the values will be plotted around zero. The distributions of differences are drawn as separate density curves on the marginal axes. We group outliers in three groups, where (a) pressure in RAOB and RAP < RTMA; (b) pressure and dewpoint in RAOB < RTMA, RAP $\approx$ RTMA; (c) temperature and dewpoint temperature in RAOB and RAP > RTMA. We keep these samples in our dataset as they reflect the true state of observations. Additionally, since each dimension in D is approximately normal, summary statistics for the mean, standard deviation and standard error are computed for each variable (Table 3.1). According to the summary statistics and distribution, we conclude that 99% of the data fall within ± 2 standard deviations of the mean, but the outliers (circled and labeled in Figure 3.6) share insightful information to the variability and analysis error.

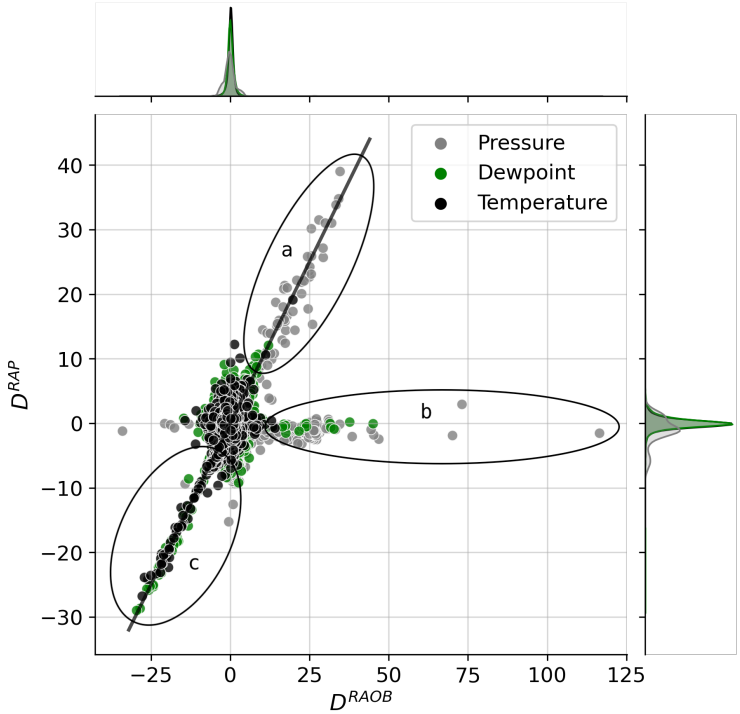


Figure 3.6: Observed differences, D, between RTMA and surface values of the RAP and RAOB for each meteorological variable of interest. The straight line represents samples where the differences D^{RAP} and D^{RAOB} are equivalent. RTMA and the surface values from the profiles agree when $D = 0$. (a) pressure in RAOB and RAP < RTMA; (b) pressure and dewpoint in RAOB < RTMA, RAP $\approx$ RTMA; (c) temperature and dewpoint temperature in RAOB and RAP > RTMA.

Table 3.1: Summary statistics for the differences of each meteorological variable between the RTMA and surface values for each variable in the RAP and RAOB profiles.

	Mean	SD	SE
D^{RAOB}			
Pressure (mb)	-0.107	2.677	0.014
Temperature (°C)	0.068	1.216	0.006
Dewpoint (°C)	0.096	1.370	0.007
D^{RAP}			
Pressure (mb)	-0.939	2.314	0.012
Temperature (°C)	-0.054	1.145	0.006
Dewpoint (°C)	0.068	1.182	0.006

Initial assumptions may lead one to believe that the RAOBs match the RTMA values more closely than the RAP estimates. However, the RTMA contains an estimate of analysis uncertainty for each meteorological variable, which reflects the background fields and spatial error correlations for the observational data [37]. In quality assessment studies of RTMA, it has been shown that the variability in the data is ± 2°C for temperature, 2–4% for relative humidity, and $\pm 0.678\,\mathrm{mb}$ for surface pressure [39, 40]. In the results above, we observe a greater magnitude of difference in the pressure measurements, which is likely a result of considering only a subset of the CONUS domain. Another reason is the temporal differences that exist in collocated data samples. That is, the RAP and RTMA data are collected hourly, and the time difference between observations may influence the accuracy of measurements as it relates to a RAOB. The inclusion of uncertainty introduces a layer of complexity when modeling the vertical profiles. In particular, a neural network will learn to identify the relationships between variables and patterns that exist in the data; however, it may overfit on what is perceived as noise during training.

3.4 Geostationary Operational Environmental Satellite

The Geostationary Operational Environmental Satellite (GOES)-R series contributes to an over 45-year history of continuous and high-resolution spatial coverage of observational imagery over

North and South America. On November 19, 2016, NOAA launched GOES-16 into geostationary orbit to replace its predecessor, GOES-13. The instruments on-board GOES-16 include an improved multi-channel passive imaging radiometer named the Advanced Baseline Imager (ABI), the Geostationary Lightning Mapper for measuring lightning activity, and a suite of other space environment sensors. The ABI serves to capture imagery of Earth's climate, weather, and environmental conditions and is the primary instrument of interest for this study.

Table 3.2: The eight channels selected from the GOES-16 ABI to use in this study.

Band Number	Central Wavelength (μm)	Nickname	Type
8	6.2	Upper-Level Tropospheric Water Vapor	IR
9	6.9	Mid-Level Tropospheric Water Vapor	IR
10	7.3	Lower-level Water Vapor	IR
11	8.4	Cloud-Top Phase	IR
13	10.3	"Clean" IR Longwave Window	IR
14	11.2	IR Longwave Window	IR
15	12.3	"Dirty" Longwave Window	IR
16	13.3	"CO_2" Longwave Infrared	IR

GOES-16 produces enormous amounts of data with scans every five minutes CONUS wide with a resolution of 0.5–$2.0\,\mathrm{km}$. The ABI uses 16 spectral bands between 0.47–$13.30\,\mu\mathrm{m}$ comprising of two visible channels, four near-infrared channels, and ten infrared channels. Individual channels are set to particular central wavelength to capture atmospheric phenomena and are used in many baseline products, such as identification of jet streams, signatures of turbulence, cloud formation and height, volcanic ash plume detection, and many others. While the ABI is not a sounder (an instrument that measures temperature and moisture as a function of height), its window and water vapor bands have some sounding capabilities. Therefore, we select channels 08-11, the water vapor bands, and 13-16 whose weighting functions primarily peak at or near the surface. We justify the use of these channels by the success of work by Schmit *et al.* [10] and their use of ABI in the legacy atmospheric profiles algorithm, as well as Hilburn [18], who demonstrates how

a subset of these channels can be used to map to vertical levels of geopotential height, temperature, and relative humidity in an NWP model. The central wavelength and descriptive meaning for each channel are described in Table 3.2.

An example image of the channel 11 ($8.4\,\mu\mathrm{m}$) converted to brightness temperature over the CONUS sector is shown in Figure 3.7. Although nicknamed the "Cloud-Top Phase" band, $8.4\,\mu\mathrm{m}$ is in a window region so there is little absorption of energy. As such, the brightness temperatures give a reasonable estimate to the surface skin temperatures, as well as the cloud-top temperatures of thick clouds.

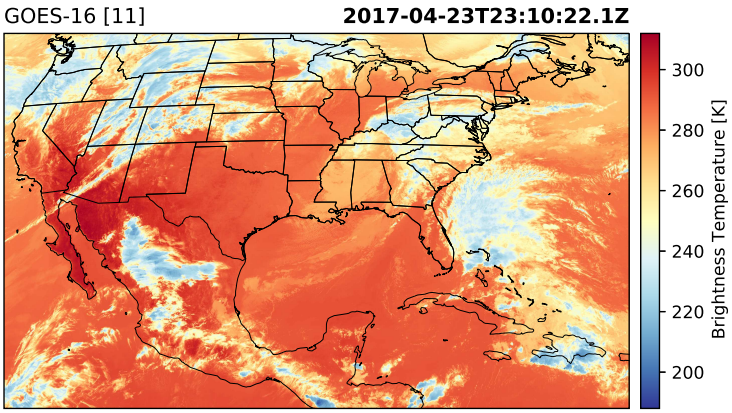


Figure 3.7: GOES-16 ABI C11 ($8.4\,\mu\mathrm{m}$) brightness temperatures over the CONUS domain.

Information from the ABI are provided from the conversion of spectral radiance to brightness temperature using the Planck function relationship and then cropped to a specific region of interest. Brightness temperature directly relates to the intensity of radiation emitted by a blackbody at a given wavelength as the temperature of that blackbody, and is the common unit of measurement for products that aid forecasters in monitoring weather, oceanographic, and environmental phenomena. We convert from radiance ($\mathrm{mW}/(\mathrm{m}^2 \cdot \mathrm{sr}^{-1} \cdot \mathrm{cm}^{-1})$) to brightness temperature (K) for each spectral band, b, with equations from the GOES-R ABI Algorithm Theoretical Basis Document [41]:

$$T_b = fk_{2,b} / \log((fk_{1,b}/L_{\lambda,b}) + 1) - bc_{1,b})/bc_{2,b}, \tag{3.8}$$

where $fk_{2,b}, fk_{1,b}, bc_{1,b}$, and $bc_{2,b}$ are Planck coefficients based on the spectral response function (SRF) of GOES-16 ABI. Calculations of $fk_{2,b}$ and $fk_{1,b}$ are made using Planck's constant, Boltzmann's constant, velocity of light, and the central wave number from the instrument. The band correction coefficients $bc_{1,b}$ and $bc_{2,b}$ are based on the intercept and slope found by regressing linear models over equally spaced radiances between the integration over the instrument's SRF and a monochromatic Planck conversion. The coefficients have a reliance on instrument and sensitivity of the sensor at different wavelengths, thus leading to unique values for each spectral band that can change with time. However, these coefficients are reported as metadata in every ABI scan, and can be directly used in (3.8).

While data volumes over the entire CONUS sector can be large, we extract a small 3×3 region of interest, which is closest in time and space to the radiosonde release time and location, from each channel. The nominal $36\,\mathrm{km}^2$ region provides an instantaneous snapshot of the environment with at most 5 minutes of separation surrounding the time of the launched radiosonde. Note that radiosondes can take up to 30 minutes to ascend, and with winds of $20\,\mathrm{m\,s}^{-1}$, neither their exact time nor location is fixed. Thereafter, we take the mean for each channel to get an average value over the area.

Chapter 4

Model Consideration

Supervised learning is the process of learning to model input-output relationships using labels or measurements as the target transformations. There exists many supervised learning algorithms, each with their own strengths and weaknesses. For this work, we leverage neural networks as a method to learn a mapping between the initial guess RAP profiles and ground truth RAOBs, due to their strengths of capturing non-linear and complex patterns in high dimensional data. To supplement the simulated environment of the RAP, we experiment with the inclusion of observational data from GOES-16 ABI (denoted GOES from hereon for brevity) and the RTMA as input to the network alongside the RAP. Specifically, we use the collocated temperature, dewpoint temperature, pressure, and geopotential height from the RAP along with the mean of each selected channel in GOES and RTMA to predict the temperature and dewpoint temperature of the RAOBs (Figure 4.1). The network architectures discussed below include references to using both sets of observation data (GOES and RTMA); however, as we discuss in Section 5.1, a sensitivity study is done to evaluate how useful the included observations are.

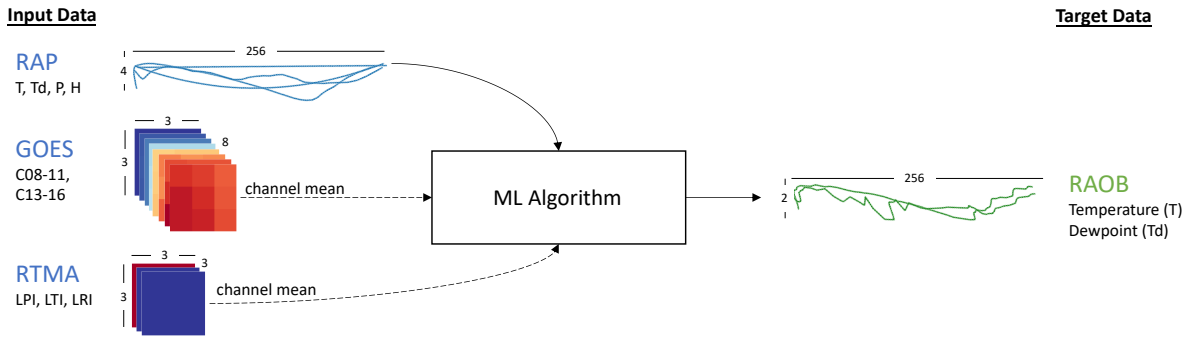


Figure 4.1: Input and output features used with the machine learning algorithm. GOES and RTMA data have dimensionality reduced to the mean of each individual channel. The dashed line represents input features that are optional.

4.1 Linear Regression

Each of the explored network architectures builds upon linear regression, which attempts to model the relationship between one or many independent and response variables by fitting a linear equation to the observed data. Given n samples with d explanatory features, $X \in \mathbb{R}^{n \times d}$, and target values with one output feature, $T \in \mathbb{R}^n$, we find a regression line defined by an affine function $g(x_i; w) = w_0 + w_1 x_{i1} + w_2 x_{i2} + \cdots + w_d x_{id}$, where w is a vector of weights with a bias inserted, for $i = 1, \ldots, n$. The model is linear in the parameters w and input x_i, which makes the model easy to solve and interpret, but greatly limits the complexity of functions the model can represent.

Extending to multiple outputs of size k, where $T \in \mathbb{R}^{n \times k}$, we randomly initialize the bias inserted weights $W \in \mathbb{R}^{k \times d+1}$ and find w_k that minimizes the error in the k^{th} output, and use it to make predictions. For our application, X represents the input features flattened into a 1-dimensional vector for each observation. That is, the RAP's four profile variables, GOES channels, and RTMA variables are vectorized and concatenated together, thereby treating all d features independently. The target samples, T, contain the temperature and dewpoint temperature profiles flattened and concatenated to k output features.

Weights in W could be solved analytically (although challenging and inefficient with high dimensional data) to minimize the squared error between the prediction Y and target T, but to be more comparable with neural networks, the weights are updated using gradient descent, which is a method that was first proposed by Augustin Cauchy [42]. First, we calculate a prediction for the i^{th} sample using $g(x_i; w_j)$ for $j = 1, \ldots, k$ to find y_i, and use this to compare with the target observation t_i. Mathematically, we can efficiently compute Y using matrix multiplication. The comparison between all samples in T and Y is preformed using a differentiable *loss function* $\mathcal{L}$, which is usually a function of the model's input and target observations that describe the error in the model. For example, using the mean-squared error (MSE) we compute the following:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{n}\frac{1}{k}\sum_{i=1}^{n}\sum_{j=1}^{k}(t_{ij} - y_{ij})^2. \tag{4.1}$$

Thereafter, incremental updates are made to the weights with gradient descent in $\mathcal{L}$ by making small changes, factored by a *learning rate* of size η, in the negative gradient direction:

$$w_k \leftarrow w_k - \eta \frac{\partial \mathcal{L}}{\partial w_k}. \tag{4.2}$$

For each update step, the function is optimized to minimize the loss using gradient descent or one of its variants, such as Stochastic Gradient Descent (SGD), Adaptive Moment Estimation (Adam), or Adaptive Gradients (AdaGrad). These optimization algorithms are alternatives that have shown to improve convergence and performance of training. Updates are typically batched using equal partitions of samples from the training data to update the weights more frequently. When updates are performed using all of the mini-batches, this constitutes a single *epoch*. Training continues for a fixed number of epochs or until the weights sufficiently converge.

4.2 Fully-Connected Neural Networks

A fully-connected neural network consists of layers of artificial neurons, whereby all outputs in one layer connect to every neuron in the subsequent layer. This class of neural networks contains no cycles and does not have any spatial context, *i.e.* spatial relationships between inputs and neurons are ignored. The depth of the network is determined by the number of *hidden layers* between the input and output layers, where each layer is comprised of one or many artificial neurons or *units*. In this thesis, we denote the number of units in each hidden layer by an array of values corresponding to each layers, *e.g.* $[5, 10]$ signifies two hidden layers with 5 units and 10 units, respectively. We show an abstract representation of this network in Figure 4.2. The network represents the target function $f(X; \Theta)$, where $\Theta = (W^{[0]}, \ldots, W^{[L-1]}, b^{[0]}, \ldots, b^{[L-1]})$ is the complete set of parameters with separate weights and biases for L layers (the sum of hidden layers and the output layer). These weights are independent of one another for every unit, and if there exists a target function f with the appropriate weights, we can satisfy $y_i = f(x_i)$ for $i = 1, \ldots, n$ with a forward pass of the

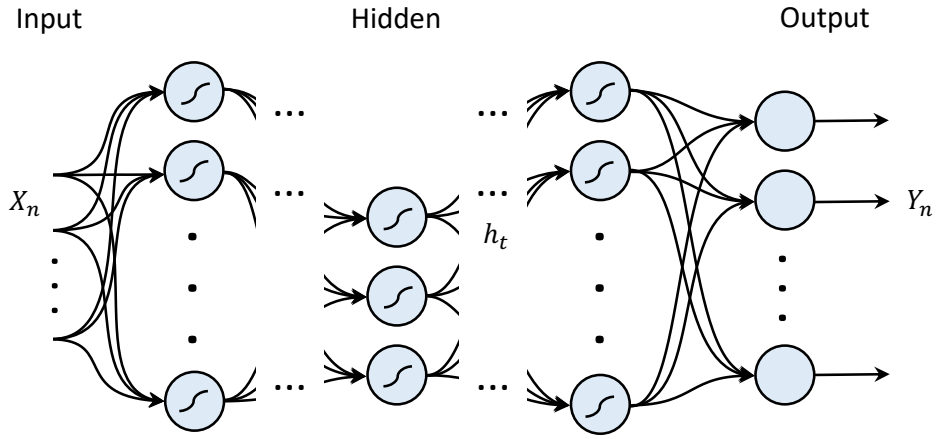


Figure 4.2: Fully-connected network architecture showing the flow of each independent feature in X_n as it propagates forward through one or many hidden layers. Each hidden layer has one or many units with non-linear activation function. The unit in the output layer is linear.

network. That is, an input sample is applied to the first hidden layer, the signal propagates forward through the remaining hidden layers, and an output is retrieved from the output layer.

Intermediate outputs from the l^{th} layer have as many output variables as the number of units in that layer. Each unit in the hidden layers performs a weighted sum of the input followed by a non-linear *activation function*. This activation function, $\sigma(\cdot)$, is traditionally sigmoidal, such as the logistic, $\sigma(x) = 1/(1 + e^{-x})$, or hyperbolic tangent, $\sigma(x) = \tanh(x)$, functions, but can also be piecewise linear with a rectified linear unit, $\sigma(x) = \text{ReLU}(x) = \max(0, x)$. Such activations in the hidden layers allow the network to learn non-linearities that exist in the data. The final output layer has a linear activation, $\sigma^{[L]}(x) = g(x)$, as the values in T are unbounded. Together, with an L-layer network, we can recursively define the forward pass of f_Θ mathematically:

$$\begin{aligned} f_\Theta^{[0]}(x) &= x, && \text{(input layer)} \\ f_\Theta^{[l]}(x) &= \sigma(W^{[l-1]} f_\Theta^{[l-1]}(x) + b^{[l-1]}) \ : \ 1 \leq l \leq L-1, && \text{(hidden layers)} \\ f_\Theta(x) &= f_\Theta^{[L]}(x) = W^{[L-1]} f_\Theta^{[L-1]}(x) + b^{[L-1]}. && \text{(output layer)} \end{aligned} \tag{4.3}$$

Weights are updated during training with gradient descent in a way similar to the linear model. We build on equation (4.2) by minimizing a differential loss function $\mathcal{L}(f_\Theta(x), t)$ using SGD, which approximates the loss over the entire training set by computing the loss over a mini-batch

of $\mathcal{B}$ samples. The gradient of $\mathcal{L}$ with respect to Θ is backpropagated to incrementally update the parameters, denoted as:

$$\Theta \leftarrow \Theta - \eta \nabla_{\Theta} \mathcal{L}_{\mathcal{B}}(f). \tag{4.4}$$

Given a fully-connected network, as depicted by the flow diagram and architecture in Figure 4.2, we need only to specify the input and output dimensions, the number of hidden units, and activation function to construct the network. Thereafter, we use an input sample and compute a forward pass of the network, through the hidden layers, and get an estimate of vertical profiles at the output. Then, we optimize a loss function to train the weights via backpropagation. After the fact, we can use the trained model to produce new estimates of soundings for samples outside of the training dataset.

4.3 Convolutional Neural Networks

With fully-connected networks, we emphasize the association that may exist among one feature and any other features by treating variables independently. As such, the network uses a significant number of parameters to learn their relation to the output, whereas Convolutional Neural Networks (CNN)s consider a neighborhood of values where nearby associations may exist. Using a weight sharing technique, CNNs are able to detect local patterns in sequence data and are spatially translation invariant. Therefore, specific patterns in a profile could be learned regardless of where it exists in the profile. This idea is useful as, for example, temperature inversions do not always exist at the same vertical level between samples or geographical locations.

A convolution is a mathematical operation that expresses the area of overlap between a spatially reversed function g as it shifts over some other function f. Specifically, the convolution of f and g is the integral over the product for two continuous functions, after one is reversed and shifted, to produce a new function, $f * g$, and is written as:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau) g(t - \tau)\, d\tau, \tag{4.5}$$

where t represents the magnitude of shift with functions indexed by τ. Additionally, we can discretize the convolution if $f, g : \mathbb{Z} \to \mathbb{R}$ and $t, \tau \in \mathbb{Z}$ by taking the sum of products for all indices:

$$(f * g)(t) = \sum_{\tau=-\infty}^{\infty} f(\tau)g(t-\tau). \tag{4.6}$$

The convolution operation can be adapted for use with finite sequences, which are often more practical for neural networks. Such sequences can be regarded as if they are infinite with zeros outside its finite range. Introducing zeros has no influence on the result of the convolution and allow for the finite sequences to overlap; therefore, the summation in equation (4.6) takes values of τ which are bounded by the extent of the kernel f, where a *kernel* represents a filter of trainable weights learned with gradient decent to extract features and patterns that exist in the profile.

We specify g to be the input sequence and f is the kernel of the convolution. The input for the first layer is the finite 1-dimensional profile containing each meteorological variable as a different channel. By convolving f and g we produce a new *tensor*, or multi-dimensional array, where the result for each index is simplified to the dot product of the kernel as it *strides*, or shifts a discrete step size, relative to the input for each channel dimension in the tensor. Increasing the number of kernels will effectively increase the number of output channels in the tensor. As with the fully-connected layers, after each convolution, we introduce non-linearity by applying an activation function to the output. Thus, for each layer we need to specify kernel size, stride size of the kernel, the number of kernels to operate over each channel, and the activation function.

Following each convolutional layer is a *pooling layer*, which works to down-sample the input by capturing the maximum or average value within a defined region. By using a stride of two we reduce the size of the tensor in half. This layer has no weights to be learned, and it is an effective method for emphasizing the most influential weights and reducing model complexity.

Figure 4.3 illustrates one of the many CNN architectures considered, which combines convolutional, pooling, and fully-connected layers to produce an estimate of the profile. The RAP is the only data used with the convolutional and pooling layers, and its output dimension depends on the depth of repeated pooling layers and the number of convolutional kernels in the last operation. The

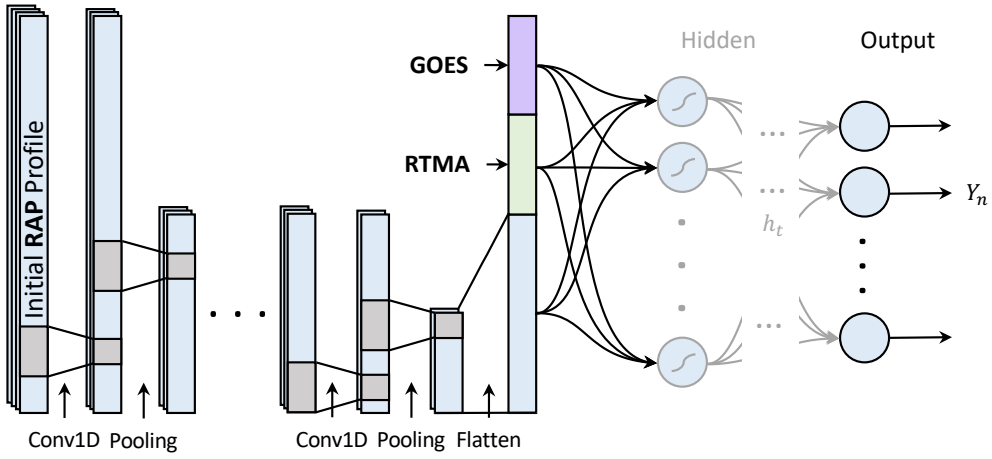


Figure 4.3: Network architecture for the convolutional neural network. Only the RAP is used as input to the convolutions and the GOES and RTMA are concatenated with the output of the last pooling layer. Zero or more hidden layers (grayed out) connect the features to the linear output layer.

output tensor is then flattened to a vector to be used as input to the following layers. The GOES and RTMA data are introduced as independent variables by concatenating a flattened vector of the observations to the flattened output vector of the convolutions. As we will see later, we can control which channels from GOES and RTMA to use or exclude as inputs to the networks. The joint vector is then used as input to zero or more hidden layers with non-linear activations and then a final linear output layer that matches the output size of the RAOB profiles.

4.4 Deep Residual U-Nets

U-Nets get their name from taking a structure similar to a convolutional autoencoder (an architecture that encodes the input through downsampling layers, and decodes the compressed representation of the input samples through a series of upsampling layers) with stacked connections of consecutive layers that take the shape of a "U". The network preserves all the advantages of CNNs with improved performance of pixelwise predictions. Figure 4.4 outlines the structure of this architecture, which includes downsampling blocks, a bottleneck layer, and upsampling blocks with joint connections to the downsamplng layers. Building upon traditional U-Nets, we include an additive connection of the input profile with the final output of the upsampling layer. Inspired

by [29], this connection forces the network to learn only the residuals between the input and target samples.

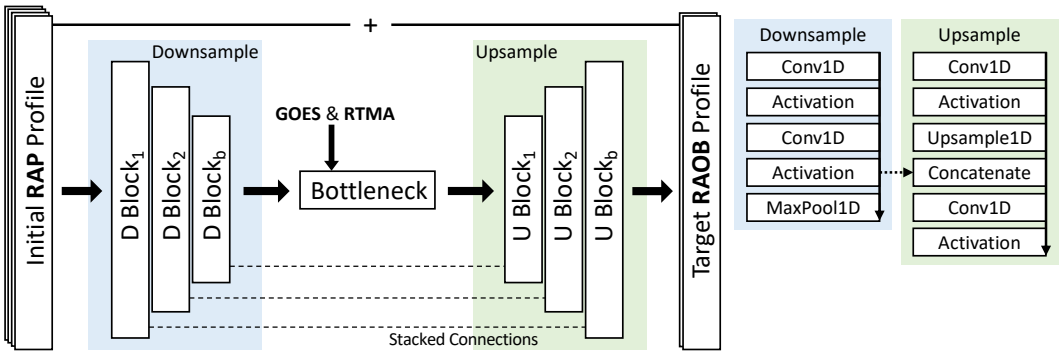


Figure 4.4: Residual U-Net architecture inspired by [29] that uses the RAP profile as input and concatenates the GOES and RTMA data in the bottleneck of network. The dotted line represents stacked connections between downsampling and upsampling blocks, and the residual connection is represented by the "+". (Optionally) following the U-Net are zero or more hidden layers and a linear output layer.

The initial RAP profile, with channels representing each meteorological variable and vertical level as individual pixels, is input to the first downsampling layer. This layer has two 1-dimensional convolutional layers, each with its own activation function, followed by a max pooling layer to reduce the size of the input in half. There are b downsampling layers before the bottleneck of the network. If there are GOES and RTMA data to use as additional predictors, then the bottleneck flattens the last downsampling layer's output and concatenates the flattened observational input features. Thereafter, the bottleneck passes the vector through a fully-connected layer and reshapes its output to match the shape of the last downsampling layer. Following are b upsampling layers, which convolve over the output from the bottleneck, upsample the output, stack the output of the b^{th} downsampling layer, and their output is used as input to another convolutional layer and activation function. The final upsampling layer has a linear activation and then the original input is added to the output of the network. This is the additive connection making it a residual network. Since the U-Net is symmetric around the bottleneck with multiple convolutional layers in each block, we simplify the notation of the network's structure, *i.e.* number of blocks and filters, by

specifying only the structure of the downsampling blocks. For example, $[32, 64, 128]$ denote three downsampling blocks and three upsampling blocks with 128 filters nearest the bottleneck.

In the downsampling layers, we learn the feature maps of the RAP profile, so why not reuse the same feature maps in the decoder of the network to convert the input to the target RAOB profile. This is one of the primary advantages of U-Nets over CNNs. As a result, we maintain the structural integrity of the RAP profile, and reduce the distortion introduced by compressing the profile to the bottleneck layer. In addition to the symmetric skip connections, the additive connection is shown in the literature to have perceptible improvements. However, with the introduction of more layers, we also have significantly more parameters, which can make it more challenging to optimize and produces results that do not overfit on the training data.

We also experiment with the inclusion of fully-connected layers following the last layer of the U-Net, although it is an optional configuration (not shown here). An illustration of this network would look similar to that of combining Figure 4.4 with Figure 4.2 at the output. This scenario flattens the output of the last layer from the U-Net and treats each variable independently as input features to one more many hidden layers. Thereafter, a linear output layer produces the final output of the network.

4.5 Reducing Overfitting

The concept of *overfitting* comes from a neural network that does not accurately reflect the data from the problem domain by memorizing the training data and failing to generalize to new samples. Specifically, when a network becomes too complex and overparameterized, it learns to fit the detail and noise in the training data, which has a negative effect on the performance of the model for unseen data. Furthermore, a network is more prone to overfit when a dataset is small in the number of samples. Shallow learning techniques can be used to overcome the issue, but also limit the complexity the target function can represent. As such, we would like to develop an architecture that minimizes overfitting and generalizes well on unseen data. We use the following techniques intermittently throughout the experiments when overfitting is observed:

- *dropout*. Dropout follows hidden and dense layers with a probability, from a Bernoulli distribution, of randomly omitting units and their connections to the successive layer during training. Since this method removes the output of units, it creates an exponential number of simpler sub-networks which share parameters. Once trained, and during inference, a forward pass through the entire network can efficiently approximate the combined predictions of all the sub-networks, effectively creating an ensemble of models and using their average. Srivastava *et al.* [43] show this technique to be effective in preventing the co-adaptation of units and reducing overfitting.

- *batch normalization*. The distribution of inputs at each layer in a network inherently change during training and can lead to instabilities and saturated non-linearity within a network. As such, batch normalization is used between the output of a layer and the activation function to normalize each scalar feature independently with zero mean and unit variance (similar to how the input data are normalized before the first layer in a network). This is done to improve gradient flow, regularization, and stabilization during training [44]. Given a mini-batch of samples $\mathcal{B}$ of size m, we follow the batch normalization algorithm for a unit's output x_i as follows:

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \tag{4.7}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \tag{4.8}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \tag{4.9}$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta, \tag{4.10}$$

  where $\hat{x}_i$ is the normalized value calculated using the mean, $\mu_{\mathcal{B}}$, variance, $\sigma^2_{\mathcal{B}}$, and constant epsilon, ϵ, over all mini-batch samples, and y_i is a linear transformation of $\hat{x}_i$ using the learned parameters γ and β. The purpose of scaling and shifting in (4.10) is to ensure the transformation added to the network can represent the transformations of an activation

function. These values are learned through backpropagation for all training steps. Therefore, the use of batch normalization introduces a number of new parameters for each output unit in a layer, namely the γ and β weights as well as the non-trainable moving mean, $\mu_{\mathcal{B}}$, and moving variance, $\sigma^2_{\mathcal{B}}$. Consequently, this technique is computationally expensive and adds to the training time of the network.

- *regularization*. Regularization allows for penalties to be applied to a layer's parameters during optimization to constrain the complexity of the model. These penalties are added to the loss function that the network optimizes during training. Two effective techniques are known as L_1- and L_2-Regularization, which have shown to provide simpler neural network solutions and reduce overfitting [45]. The L_1 penalty, $||\Theta||_1$, aims to push the parameters toward zero by taking the sum of the absolute value of the parameters. This term is multiplied by a hyperparameter factor of λ and added to the loss as $\mathcal{L}_1 = \mathcal{L}_{\mathcal{B}}(f) + \lambda||\Theta||_1$. A higher value of λ will bias the selection to models of lower complexity, though too large a value may lead the model to underfit. By forcing parameters toward zero, we increase sparsity and reduce variable importance selectively through training, which assists in the network building a representation of important features. Alternatively, the L_2 penalty, $||\Theta||_2^2$, forces weights to be small and non-zero values by minimizing the sum of the squared magnitude, but is not as robust to outliers as square terms are emphasizes. Adding to the loss function, we have $\mathcal{L}_2 = \mathcal{L}_{\mathcal{B}}(f) + \lambda||\Theta||_2^2$. Similarly, we use λ to control how much regularization the model should have.

Chapter 5

Experimental Setup

This chapter begins by discussing the dataset partitions and particular data features that are used during experimentation and evaluation. Thereafter, we describe the general overview for how we setup the models and configure their hyperparameters. Lastly, we outline multiple loss functions, and introduce a knowledge guided loss that incorporates derived indices from the profile.

5.1 Experimental Procedure

An elaborate architecture search, hyperparameter optimization, and data information analysis is conducted and described in detail below. The motivation behind exploring different neural network architectures is to understand the degree of complexity (in terms of the network's connections, individual operations, and number of parameters) that is needed to model the relationship between the RAP and RAOBs. Starting with a linear model we get a sense of the baseline linear correlations of predictor variables. Since the RAP profile is used as input, we expect that the output of the linear model is no worse than the initial guess. By exploring the other models outlined in Chapter 4, we can better understand the non-linear and spatial correlations of the predictors. However, the choice of architecture and its hyperparameters are only a part of setting up the experiments. There also exist a number of ways in which we utilize the data that can influence the outcome of our results.

5.1.1 Data Utilization

The preprocessed and collocated data are partitioned spatiotemporally and shuffled into a training (75%), validation (10%) and test (15%) sets. This is done to reduce bias in any one dataset by equally distributing the sample launch locations and release times among the partitions. Moreover, the soundings have low autocorrelation and are naturally separated due to the temporal and spatial sparsity of samples, thus allowing data to be shuffled equally among partitions. Effectively, the training data contains $28,782$ samples and is used only to train and update the parameters of the

network. The validation data is a subset of the data held to give an unbiased estimate of error in the model during training. After every epoch we compute statistical metrics over the training and validation set and use this information to control training and quantify performance. The test set is reserved for after training to evaluate and make fair comparisons of separate models.

For each network architecture we use as input the RAP with the RAOB data as the target output. Additionally, we (optionally) introduce the RTMA and GOES data as input, either individually or together. Doing so leaves four potential combinations to use as input: (a) RAP, (b) RAP+GOES, (c) RAP+RTMA, and (d) RAP+GOES+RTMA. Performing a comparison of the different combinations allows for the assessment over the observational data and their significance to improve predictions. The caveat to this analysis is that when more features are introduced to the network there is an associated growth in the number of parameters. In some situations, for example, when using the convolutional or U-Net architecture, the preserved spatial information is lost as fully-connected layers are needed to concatenate the additional observational measurements. Thus, the choice of architecture is restricted when using observation data and models that are fully-convolutional can not be used as the observational features need to be joined independently.

In addition to partitioning the data and separating input variables for different studies, we also label samples as either cloudy or clear-sky based on the observed conditions from the GOES-R L2+ Clear Sky Mask [46]. The mask provides binary classification for each pixel using the GOES-R ABI visible, near-infrared and infrared bands. As such, the mask has the same grid spacing and resolution as the GOES-16 ABI data. We decide to identify a sample as cloudy if a $100\,\mathrm{km}^2$ region, collocated over the radiosonde observation, has more than 85% of the pixels labeled as cloudy. We use this slightly larger region, as compared to the GOES ABI data, to account for radiosonde drift and elapsed time of the cloud movement. The primary reason to consider cloud coverage is two-fold, that is (a) we can train separate models using all of the samples or only cloudy/clear-sky samples and (b) we can evaluate a model trained using all data samples under either condition. The general profile accuracy and architecture search considers both clear- and cloudy-conditions. Separation of the data makes the most sense meteorologically when using satellite radiances as

input. This is because clear-sky radiances are not obstructed or absorbed by any clouds, thereby capturing a better representation of the atmosphere at different wavelengths. However, we reduce our effective sample size at the expense of data separation, which is generally detrimental to neural network performance. Additional details and results are outlined in Section 7.1.

Regardless of the data combination or features used, and before training our models, we standardize the input and target variables to have a mean of zero and unit variance (z-score normalization). Every vertical level of the profile and each observational variable are standardized independently by subtracting the mean and dividing by the standard deviation from the training data. To convert predictions from a model back to their original units, we simply multiply by the standard deviation and add the mean using the statistical values from the RAOB. While the networks have no assumptions about the underlying distribution of the data, this is a crucial step before both training and inference as to not saturate the hidden units and to maintain relative scaling of features.

5.1.2 Model Setup and Hyperparameters

As a way to assess the feasibility of neural networks in this work, we explore the search space of different network architectures and hyperparameters. The *hyperparameters* are the non-trainable parameters which describe the topology of a network (*e.g.*, number of hidden layers and units) and the parameters used to train the model (*e.g.*, learning rate and activation functions). These parameters are initialized prior to optimizing the model and do not change. Through experimentation we constrain the infinite combinations of hyperparameters to the parameters that demonstrate the best performance with recurring patterns. For each model setup we train five separate models with different initial weights to capture a comprehensive understanding of that particular model. It is possible that a model with initial parameters furthest from the data mapping of the input and output will fall into a local minimum and fail to converge to better optimized values. Conversely, a model initialized with more appropriate weights may appear to perform better as the parameters converge to values that yield a more optimal loss. Therefore, running multiple trials will increase

confidence in the performance of a model when making comparisons to other model setups. When investigating an individual model, we take the best of the trials and report the appropriate statistics.

All network variants consider shallow and deep structures with a narrow and wide number of hidden units and filters in each layer. In the fully-connected networks we vary the number of layers between one and five, where each layer has between 2^2 through 2^{10} units. The convolutional networks have between three and six convolutional layers with the number of filters ranging from 2^4 and 2^9 following an exponential increase with depth. More specifically, as depth of the convolutions layers increases so does the number of filters. This technique is commonly employed in convolutional networks in computer vision as early layers can identify higher level features in the data, and the later layers can identify more fine grain details. Therefore, we need not have a large number of filters in early layers, and we can add more to the subsequent layers to capture the small changes. Following the convolutional layers, we stack fully-connected layers with zero to two hidden layers of $[256]$ and $[512, 256]$ units. The observational data from RTMA and GOES are concatenated prior to passing through these layers. We experiment with a range of 2^4 and 2^8 filters with one through four downsampling and upsampling blocks. Similar to the convolutional network, the U-Net increases in the number of filters nearest to the bottleneck layer and we stack zero to two hidden layers after the last convolutional layer. Note, use of $[0]$ with the U-Net represents only the output from the U-Net.

Given a structure defining the topology of a network we then outline the parameters and operations used for training. Following each hidden layer is a non-linear activation function. The hyperbolic tangent and rectified linear unit (ReLU) are used in initial experiments, and since we found negligible differences, we use ReLU in the remaining experiments. Similarly, with optimization functions, we find adaptive moment estimation (Adam) to be sufficient in initial experiments. We initialize Adam with a learning rate η and coefficients β_n to tune how quickly and accurately the models will learn. Various values of η between $1e^{-}2$ and $1e^{-}4$ are used to determine which contribution of the gradient is most effective for the data being used. Additionally, we initialize

each coefficient to its default values for computing running averages of the gradient and its square with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, respectively.

Individual models are developed using TensorFlow version $2.3.1$ and trained on a single 24 GB NVIDIA Quadro RTX 6000 GPU with two Intel Xeon Silver 4216 CPUs at 2.10 GHz and 256 GB DDR4 memory. Having a large amount of memory on the system and GPU allow us to test great breadth of models in timely manner. However, as with most research in machine learning, it is not possible to explore the entire search space. On average we find the models to complete an epoch in four seconds and train a model in the range of a few minutes.

5.2 Loss Functions

A natural characteristic of the RAP's profiles is the increase in error at higher altitudes for which both profiles on average have their highest errors at $17\,\mathrm{km}$ above the surface. Their average minimum errors differ with the moisture profile having an error of $1.238°$C directly at the surface, and the temperature profile with an average minimum error of $0.713°$C roughly $8\,\mathrm{km}$ above the surface. The challenge with traditional loss functions, such as the mean-absolute error (MAE) and MSE, is the emphasis on large deviations of outputs with no regard to spatial context. While it is important to correct for these large errors, it is also critical to minimize the errors near the surface. The loss function being optimized is a custom weighted mean-absolute error (WMAE), which builds upon the MAE. The (MAE) shown in (5.1) computes the mean over the absolute difference of the network's output and target variables for a mini-batch of samples at every vertical level, given by:

$$\mathcal{L}_{\text{MAE}} = \frac{1}{n}\frac{1}{k}\sum_{i=1}^{n}\sum_{j=1}^{k}|t_{ij} - y_{ij}|. \tag{5.1}$$

This function is minimized by the conditional median and is mathematically more resilient to outliers when compared to MSE. Since the errors in the RAP profiles are most significant at higher altitude the MSE will bias the network to correct for these values. However, we are more focused on improving the region between the surface and $2.5\,\mathrm{km}$ above the surface. Therefore, we find MAE to be a more suitable starting point to remove bias of high differences and we introduce a

weighting factor to individual outputs based on their vertical level. Output features are consistent with altitude, but not necessarily with atmospheric conditions including cloud height, temperature inversions, convective condensation level, etc. Therefore, we assign output weights in WMAE as a function of altitude where values near the surface have a greater magnitude of error. The WMAE is mathematically represented as:

$$\mathcal{L}_{\text{WMAE}} = \frac{1}{n}\frac{1}{k}\sum_{i=1}^{n}\sum_{j=1}^{k}(\alpha \exp(-\lambda j) + \beta)|t_{ij} - y_{ij}|, \tag{5.2}$$

where α is an initial value, λ is a decay constant, and β is an offset value for an exponential decay function. The absolute difference of the profile is multiplied by this function such that the difference decays with altitude. Through initial experiments of fully-connected and convolutional networks, we find $\alpha = 3.75$, $\lambda = 0.01$, and $\beta = 0.25$ to be appropriate values for the data.

The aforementioned loss functions are standard in the computer science literature with the exception of WMAE, which we adapt to emphasize features that are important meteorologically. We extend on the idea of including domain knowledge in training by leveraging information derived from the profiles into a loss function. From a given sounding, the total precipitable water (TPW) can be calculated to describe the depth of water in a column of the atmosphere if all of the water vapor were condensed. Forecasters use TPW to know how much moisture in the column could potentially precipitate as rain or snow. As is, the accuracy of the derived product is essential to accurately predict and track severe weather events. Using this knowledge, we include an additional term in the loss that not only minimizes the error for every vertical level (with MAE), but also minimizes the error in TPW derived from the network estimate and the target RAOB profiles. Mathematically, TPW is found by integrating the atmospheric column moisture with the equation from [47]:

$$TPW = \frac{1}{\rho_w g}\int_{p_{sfc}}^{p_{top}} r\, dp, \tag{5.3}$$

where ρ_w is the water density, $999.975\,\mathrm{kg\,m^{-3}}$, g is the gravitational constant, $9.807\,\mathrm{m\,s^{-2}}$, r is the mixing ratio (dimensionless) of water vapor in mb at pressure level p, and p_{sfc} and p_{top} are the

surface and upper-level air pressure in mb, respectively. To find r we first compute the saturation vapor (partial) pressure using the following formula from [48]:

$$e = 6.112 \exp\left(\frac{17.67\, t_d}{t_d + 243.5}\right), \tag{5.4}$$

for the dewpoint temperature, t_d, in °C. Thereafter we find the mixing ratio, r, given its partial pressure, e, and the total pressure, p, of the air using the equation from [49]:

$$r = \epsilon \frac{e}{p - e}, \tag{5.5}$$

where $\epsilon = m_w/m_d$ is the ratio of the molecular weight of water vapor, $m_w = 18.015\,\mathrm{g\,mol^{-1}}$, to dry air, $m_d = 28.966\,\mathrm{g\,mol^{-1}}$. Plugging r into (5.3) gives a single quantity value, converted to millimeters of TPW over the profile. Thus, after each training step, we minimize the MSE of the derived TPW from the model estimates and target profiles in addition to the MAE of all output features (from (5.1)):

$$\begin{aligned} \mathcal{L}_{\text{TPW}} &= \frac{1}{n}\sum_{i=1}^{n}(\text{TPW}_{t_i} - \text{TPW}_{y_i})^2, \\ \mathcal{L}_{\text{TMAE}} &= \mathcal{L}_{\text{MAE}} + \alpha \mathcal{L}_{\text{TPW}}. \end{aligned} \tag{5.6}$$

In our experiments, we found that the contribution of $\mathcal{L}_{\text{TPW}}$ can be large relative to $\mathcal{L}_{\text{MAE}}$, and using a weighting factor of $\alpha = 0.25$ for $\mathcal{L}_{\text{TPW}}$ stabilizes training and produces better results. The computation of $\mathcal{L}_{\text{TPW}}$ requires the dewpoint profile, which we get from the estimate and target profiles, and the total pressure from the sounding. Thus, we use the target pressure measurements from the i^{th} RAOB to compute TPW_{t_i} and TPW_{y_i}. Including this additional argument in a loss function is particularly difficult under TensorFlow's implementation. Ideally, we would like to include a pointer to the vector of pressure values with an index to the respective sample so that we can compute TPW for each sample during training, but this is not possible given the current version of TensorFlow. To include the associated pressure in the loss, we include an additional input of RAOB pressure to the network and concatenate that input to the output. Thereafter, we extract

the pressure profiles from the network's prediction, unstandardize the dewpoint profile using the means and standard deviations of the RAOBs, and then compute TPW for the estimate and target profiles. Note that the calculation of $\mathcal{L}_{\text{MAE}}$ is on the standardized output data.

Learning curves provide an overview of how well the network learns and generalizes to unseen data by evaluating the loss over the training and validation data during training. By observing the two curves, certain properties such as the convergence in a minima of the loss (when training plateaus) and overfitting in a model (when validation and training loss diverge) can be observed. Overfitting is seen when the training loss continues to trend downward and the validation increases or stays constant. As a way to stop a model from training unnecessarily long and to reduce overfitting, a concept known as early stopping is used. This monitoring function stops the network from training at the point of smallest loss with respect to the validation data. The definition of this particular moment is when the absolute change of the validation loss is less than 0.001 for more than 10 epochs. All models employ early stopping to improve performance, which in turn, causes models to train for a different number of epochs. To encourage the use of early stopping we empirically set the number of total possible epochs to 200, which is not reached by any networks during training.

5.3 Metrics

The primary way to evaluate network performance and profile accuracy is with the root-mean-squared error (RMSE) between the estimates and target profiles. This performance metric is common in the literature when comparisons are made with RAOBs. In this work, we use the RMSE in a number of different ways, which offers a high level to fine grain representation of accuracy. Since the output of the network is two 1-dimensional profiles, we can compute the error over every single output feature or for the two profiles independently. Quantifying the error over all outputs renders a complete view for both profiles, but may include bias toward a given profile. For example, the model may produce relatively low errors in the dewpoint profile and not the temperature profile, which in turn will yield a low error but bias the dewpoint profile. Thus, we consider the

error for the two profiles independently as separate metrics. Moreover, we extend our analysis over each profile by computing the RMSE for every vertical level by mathematically taking the mean at each level separately. Doing so provides more fine grain insight to how accurate each profile is and where improvements are being made. Lastly, with concern of accuracy near the surface, defined by the first 1.5–$2.0\,\mathrm{km}$, we additionally assess the error of the first 25 levels ($1.66\,\mathrm{km}$) for both profiles and independently.

In addition to standard accuracy measures of profile measurements we also consider accuracy of radiosonde specific products computed from estimates as compared to the products generated from the RAOBs. In meteorology, a measure of convective available potential energy (CAPE) and convective inhibition (CIN) are standard indicators of convective instability found from the temperature and moisture profiles in a sounding. Additional details on these derived indices and their calculations are discussed in Chapter 8; however, at a high level, if improvements are made to the RAP profiles, then ideally, the values of derived products will be closer to those of the RAOBs as well. Therefore, we include the coefficient of determination, denoted R^2, and the RMSE of CAPE and CIN values between the estimates and RAOBs as an additional metric. Generally, R^2 represents the proportion of variance in a dependent variable that is explained by the independent features. We formulate this coefficient using the fraction of sum of squares of residuals and the total sum of squares, $\mathrm{R}^2 = 1 - \sum_{i=1}^{n}(t_i - y_i)^2 / \sum_{i=1}^{n}(t_i - \bar{t})$, for $i = 1, \ldots, n$ samples where t and y are the CAPE or CIN for the RAOB and RAP/ML estimate, respectively. A value of zero represents a model that explains no variance, and conversely, a value of one would have all the observed variation explained by the model's input. Both of these metrics are relative to the baseline metrics of CAPE and CIN values in the RAP and RAOB. Thus, an improvement to the RAP's derived indices is seen when the estimated metrics outperform the baseline metrics.

Chapter 6

Choosing a Network Architecture

This chapter explores the efficacy of machine learning in improving vertical profiles. It first looks at different neural network architectures and their general performance under different configurations. Results for the general model performance use all available data (partitioned into training, validation and test sets) to train a collection of different models. For consistency, the initial search does not include methods to reduce overfitting. Thereafter, models found to overfit are retrained using a combination of different techniques discussed in Section 4.5 and the top performing models are reported. The following sections are broken up to provide individual comparisons of each architecture. For brevity, the results of the linear model are included in Section 6.1 with fully-connected structures. Lastly, within Section 6.5 we summarize the best model from each architecture and determine the most ideal structure and specific model parameters.

6.1 Fully Connected Networks

The depth and width of the fully-connected networks have a significant impact on performance. The *depth* is defined by the number of layers and the number of units in each layer constitutes the *width*. Figure 6.1 illustrates the mean and standard error of the RMSE over all output features (the vertical levels in the two profiles) for the test set with changes in the structure. Regardless of the input variables, the total profile errors generally decrease as the width of the network increases. An increase of depth in narrow networks (with a width of ≤ 64 units) is not particularly beneficial in improving accuracy, in fact it makes the errors worse. However, networks containing more than 128 units are seen with better estimates when using two or three hidden layers compared to just one, four, or five layers. Adding more than three layers in these wider networks is disadvantageous as the network starts to overfit and learn spurious features and the noise in the training data. Thus, in general, it is beneficial to balance the depth with fewer layers and increase the width of the network for this application.

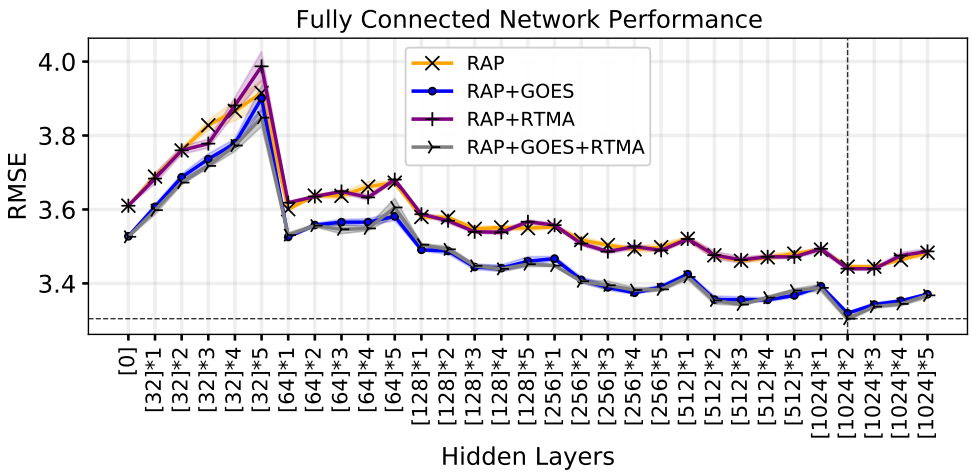


Figure 6.1: Root-Mean-Squared Error is computed for each network architecture over all output features in the test set. Network architecture is described by the x-axis (*e.g.*, $[0]$ is linear and $[64]*3$ represents a network with three hidden layers with 64 units each). Each network is trained five times and every point represents the mean error and is shaded by the standard error (barely visible in this figure). The legend denotes the results when using as input the RAP; RAP and GOES; RAP and RTMA; RAP, GOES, and RTMA.

Using different combinations of data as input reveals the GOES brightness temperatures add helpful information for learning the profiles. Conversely, including the RTMA with the RAP as input offers no additional information with fully-connected networks as mean RMSE falls within the standard error of the models using the RAP alone. A similar behavior is seen when using the RAP+GOES+RTMA as input. The mean output error of models including GOES and RTMA is not significantly different to the models using GOES. A possible explanation as to why the RTMA is not particularly useful is the inherent variability in the data, which we discuss in Section 3.3.1. As a result, there are no consistent patterns for the network to learn and the additional features are not particularly useful.

All of the models in the initial network search are trained without using any of the described techniques to reduce overfitting. However, the learning curves of these standard networks begin to diverge with an increase in magnitude following the depth of the network. Figure 6.2 outlines this behavior in a series of plots, each with a different number of layers, and compares the standard model to those trained to reduce overfitting. In particular, we retrain models using a combination of dropout with a probability of 0.20, batch normalization with default parameters, and L_2 kernel

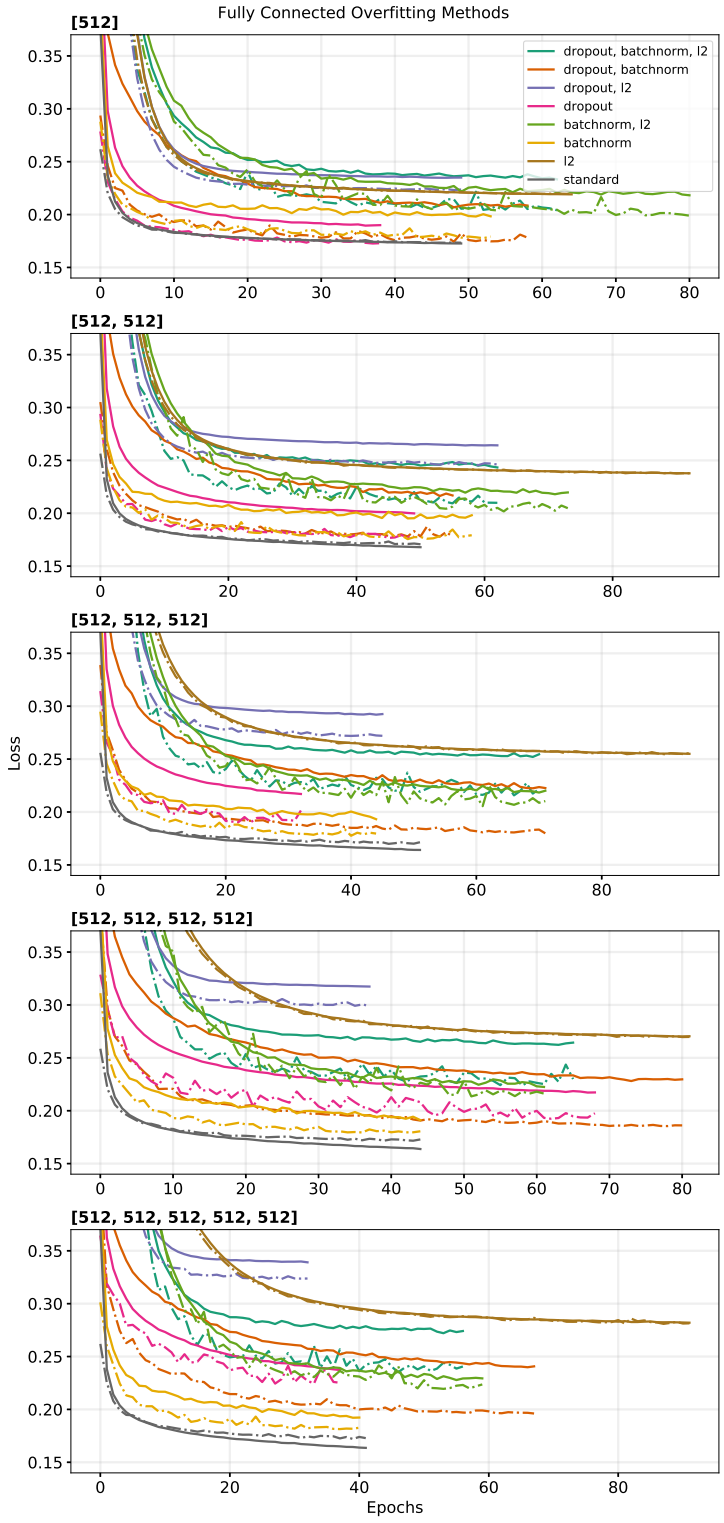


Figure 6.2: Five different network architectures of increasing depth from one to five layers with the WMAE loss (y-axis) plotted during training epochs (x-axis). Each network uses a different technique to reduce overfitting, and the standard model uses none of these techniques. Solid lines represent the loss on training data and dashed lines are for the validation data.

regularization having a regularization factor of 0.001. One immediate observation is that each model is trained for a different number of epochs. This response is intentional to illustrate the workings of early stopping; although, similar, if not more severe, results are seen even when early stopping is not used (not shown here) – since the validation loss continues to diverge from training.

The validation loss in every combination falls below the training loss, which indicates that the models are no longer overfitting, but they all have a greater loss than the standard model. Additionally, the metrics on the test set (not shown here) confirm that the use of overfitting techniques do not improve performance with fully-connected networks. The most similar case to the standard model is using batch normalization, either by itself or together with dropout, which does not experience overfitting, but still performs slightly worse. Moreover, the drawback of using batch normalization, aside from performance, is the increase in training time as additional computations are needed to transform variables and store running statistics.

Summary: The deeper networks lack generalization that traditional overfitting techniques cannot combat. As a result, using a more shallow network with enough hidden units to capture patterns and then training for fewer epochs with the RAP and GOES as input is the best solution for fully-connected networks. However, when using a linear or fully-connected network, we observe the output profiles to be jagged and contain noise between vertical levels.

6.2 Convolutional Networks

Convolutional neural networks are evaluated in a similar manner to fully-connected networks. Using different combinations of input data we train a number of network architectures and measure the test error over all output features and those near the surface. Figure 6.3 illustrates these metrics by grouping the convolutional structure with three different fully-connected layers ($[0]$, $[256]$, and $[512, 256]$) following the last convolutional layer. From this figure we see recurring patterns in the errors of the networks and the use of fully-connected layers.

The results in Figure 6.3 show the model with the smallest error containing five convolutional blocks of size $[32, 64, 128, 256, 512]$ each using a filter size of 3×1 with a stride size of one,

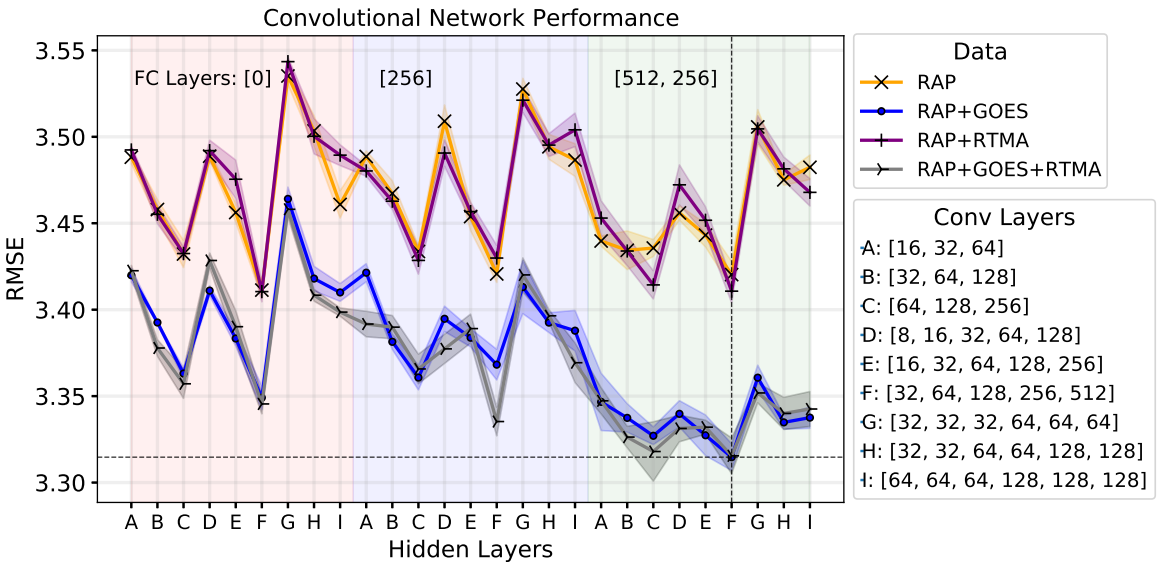


Figure 6.3: Root-Mean-Squared Error is computed for each network architecture over all output features in the test set. Network architecture is described by the x-axis and associated legend. Background shading groups the fully-connected structure that is used after the convolutions. Each network is trained five times and every point represents the mean error and is shaded by the standard error.

followed by two fully-connected layers of size $[512, 256]$, and a final output layer. In each of the three groups of fully-connected layers, the same convolutional structure, referred hereon as F_{conv}, has the lowest error. When compared to the three other five layer convolutional structures it becomes clear that ascending the number of filters in the latter layers is beneficial. Traditionally, CNNs with more filters allow the network to extract more abstractions from the data. In early layers, the primitive regularities in the data can be identified with fewer filters, but as the patterns get more complex in subsequent layers, having more filters can capture the larger combinations of patterns. This behavior is well understood and implemented in most 1- and 2-dimensional convolutional networks. As a result, we use this intuition to understand why F_{conv} outperforms the other convolutional architectures.

The use of fully-connected layers after the convolutions is to capture non-linear relationships between the abstractions found in the RAP and the observational data from GOES and RTMA. Fully-connected layers in the case when only the RAP data is input to the network attempt to learn any non-linear relationships between the individual abstractions. However, our results show no

benefit of using these layers as there are no changes to the network's overall error on average. Similar to the fully-connected networks, when introducing the RTMA, the networks perform no better than those trained with only the RAP and the mean errors typically fall in the range of standard error. Additionally, the change in error when using fully-connected layers with the RAP and RTMA as input remain consistent with changes in the number of convolutional layers. Alternatively, the inclusion of GOES show a more compelling result with lower profile errors. Not only does this suggest a linear relationship between the RAP abstractions and GOES, which improves performance, but adding non-linear units before the output improves overall accuracy even more. This observation is seen in Figure 6.3 with the divergence of profile errors from the models using only the RAP as input.

Errors near the surface (not shown here) contain patterns specific to the network structure, but the variability of performance among the models is relatively small. For example, the simpler three layer networks tend to have lower errors, and the difference with complex network structures such as F_{conv} is within ± 0.02°C of its mean error. As such, we prioritize the overall error when selecting the best convolutional network. Another relevant observation of near-surface errors show negligible differences in the use of observational data. The standard error is greater over each trial and there is overlap between the mean and standard errors of models using other combinations of the input.

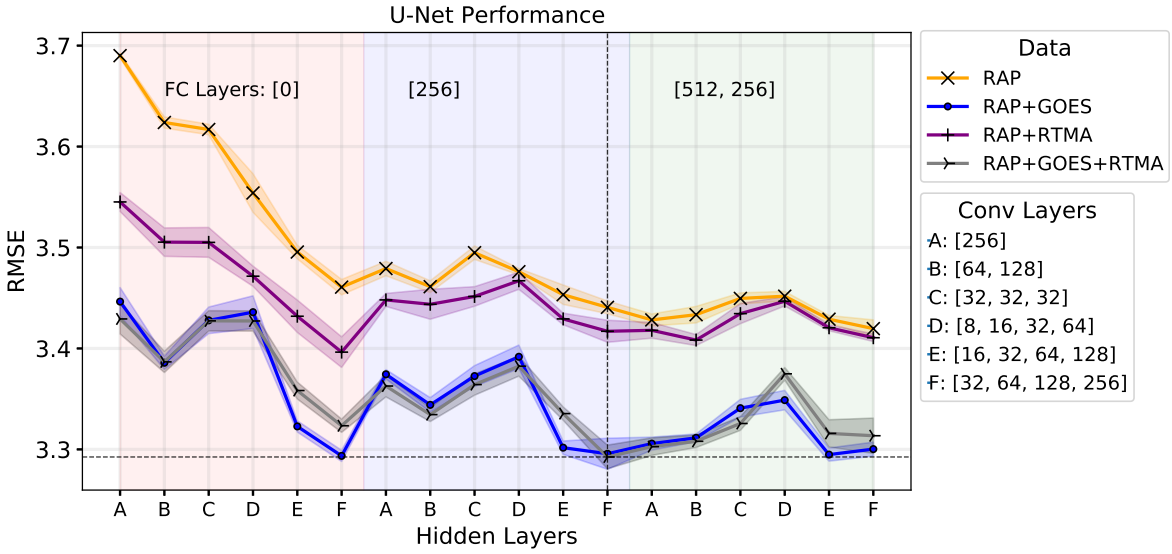
Summary: The best convolutional network structure is found having a large number of convolutions layers, which have an increase of filters with depth, and then appending two non-linear fully-connected layers after the convolutions. Results show the use of fully-connected layers to generally improve performance when using the RAP and GOES as input, and there is no advantage of using the RTMA data. Output profiles are relatively smooth, compared to the fully-connected networks, and suggest that spatial information is better captured when convolving over the RAP data.

6.3 Residual U-Net

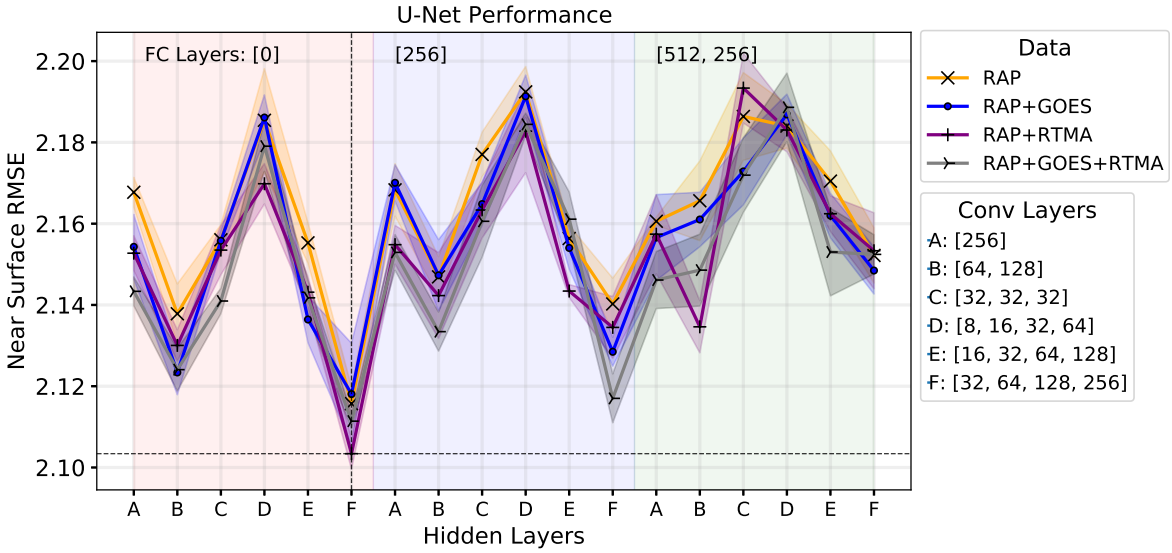
As an alternative to fully-connected and convolutional networks, the U-Net model can be configured to be fully convolutional, *i.e.* not using any fully-connected layers, when using only the RAP as input. Fully-connected layers are added only when concatenating observational data or when appending additional layers at the end of the network. However, these additional layers following the U-Net are found to be unnecessary, and the standard architecture with a bottleneck layer sufficiently improves the accuracy of the model. A complete summary of these models as displayed in Figure 6.4 validate this behavior.

Including RTMA as input changes the structure of the network by requiring an additional input layer and concatenation with the encoded RAP data in the bottleneck. This architectural change with the observational data shows an improvement to overall performance when compared to only using the RAP as input (Figure 6.4a). Comparing networks that include RTMA with those including GOES more effectively shows the benefit of satellite radiances, which is an observation seen in earlier network considerations, and suggests that RTMA is not useful. Furthermore, including both GOES and RTMA show no significant improvements over the models that include only GOES with the RAP. In fact, some scenarios using only the RAP and GOES have lower RMSE values over models using both GOES and RTMA with the RAP. This is likely as a result of the high variability in the difference of RTMA and surface values of vertical profiles. However, it is particularly challenging to fully understand why and how particular variables are used. Even by observing the weights in the network, a complete picture as to why the network learns a specific pattern or attributes importance to specific relationships is not explicit in this study. This challenge is something we discuss in more detail later on.

Near the surface, the networks following the standard U-Net architecture (without subsequent fully-connected layers, as denoted by $[0]$) are more stable and produce the lowest errors. As fully-connected layers are added, the network produces higher mean errors with a greater standard error, which indicate the sample of trials are widely spread around the true model error and the networks are unstable when initialized with different random weights. Figure 6.4b illustrates the errors



(a) Error in the total output features.



(b) Error in the near-surface output features.

Figure 6.4: Root-Mean-Squared Error is computed over the test set for each network architecture over (a) all output features and (b) only the near-surface (from the surface to $1667.5\,\mathrm{km}$) output features. Network architecture is described by the x-axis and associated legend. Background shading groups the fully-connected structure that is used after the U-Net architecture. Each network is trained five times and every point represents the mean error and is shaded by the standard error.

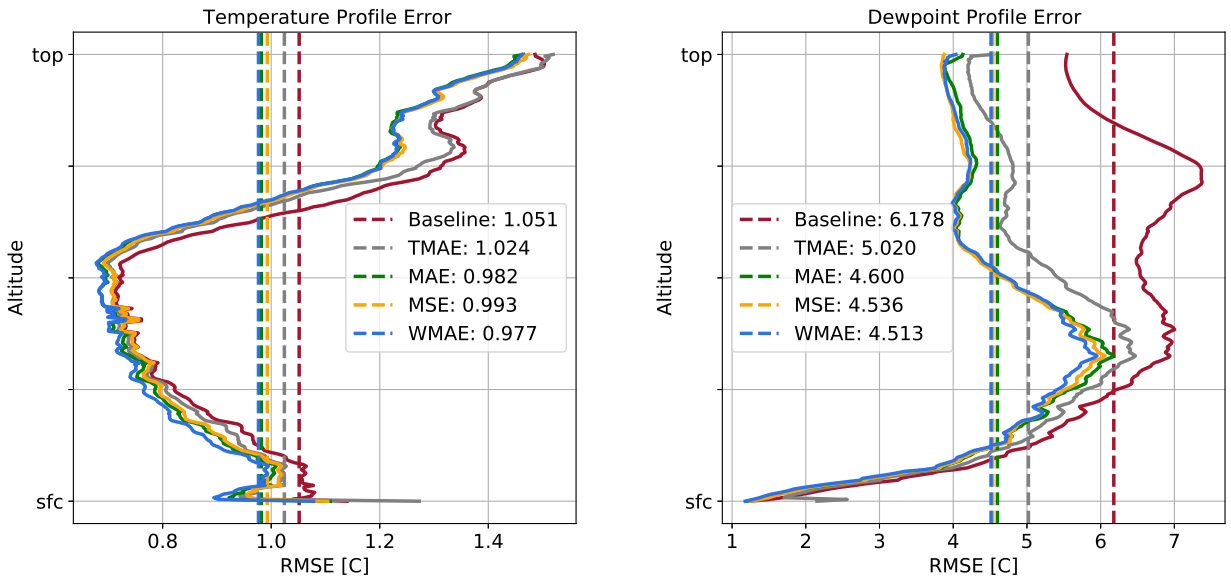
near the surface and the result of varying architectures. Here we see the network structure of size $[32, 64, 128, 256]$, with a mirrored design around the bottleneck and no additional layers after the U-Net, using the RAP and RTMA as input resulting in the minimum observed mean near-surface error. However, we find that the use of RTMA as input to this structure to have a wide spread of errors and to have worse overall errors, thus we exclude the use of RTMA in the remaining experiments unless otherwise noted. This particular network does not have the lowest error over all output features, relative to some of the other U-Net architectures, but we find the stability of near surface accuracy and general performance to be more sufficient.

Summary: The Residual U-Net architecture performs best when using more convolutional layers with stacked connections and no fully-connected layers after the output of the U-Net. Results show the best model to use the RAP and GOES as input to the network with $[32, 64, 128, 256]$ convolutional layers mirrored around the bottleneck (where the GOES is added as input). We use this network for the remainder of our experiments unless otherwise noted.

6.4 Contrasting Loss Functions

The loss function being optimized in neural networks are a critical formulation in the design of accurate networks. Here we contrast the results of four different loss functions, namely: MSE, MAE, WMAE, and TMAE. Both MSE and MAE are standard loss functions for regression problems in machine learning and both WMAE and TMAE were designed specifically for this problem. Rather than computing general metrics over the entire profile, we compute the RMSE at every vertical level for all samples in the test set for comparison. As such, we train four separate models for the four different loss functions and display the results in Figure 6.5 with a comparison of the RAP baseline error in each profile.

The loss that performs the worst is the TMAE, which includes a term to minimize the error in TPW of the ML estimate. The resulting errors at every vertical level are significantly higher than that of the three other loss functions, and the errors near the surface are worse than the baseline RAP. In our experiments, the network trained with TMAE begins to overfit on the training data



(a) Temperature

(b) Dewpoint

Figure 6.5: Networks trained with different loss function showing their errors in the test set at every vertical level as compared to the baseline error between the RAP and RAOB. The vertical dashed lines and respective numbers in the legend represent the RMSE over all vertical levels and samples. (a) Temperature profile errors. (b) Dewpoint profile errors.

much earlier and more severely than the other methods. As previous discussed in Section 5.2, we include a weighting factor to reduce the contribution of TPW errors, which helps to reduce overfitting, but as expected, begins to approach the MAE. We believe there to be potential in using a knowledge guided loss function for this application, but find TMAE to be inadequate in our experiments. As possible future work, we outline a loss which incorporates additional information about the surface, which could be helpful, in Section 9.4.

Differences between the other three models are small yet evident at specific altitudes. Models trained with MSE have a tendency to correct for errors with high magnitude and have less emphasis on outputs with low errors. In Figure 6.5a we see the MSE model does no better than the RAP where the original error is lowest, whereas MAE and WMAE provide general improvements in this region and yield lower overall errors. The values near the surface show where MAE and WMAE are most different. That is, by adding more weight to the absolute difference near the surface

we see a decrease of error. Thus, WMAE is used hereon for the remaining experiments unless otherwise specified.

6.5 Summary of Profile Accuracy

Profile and near-surface errors are presented in Table 6.1 with changes to the input test data. Here we show the best performing network from the five trials of the same architecture initialized with different weights for the four different architectures. The change in results between experiments have consistent yet marginal numerical differences, albeit small changes represent considerable improvements when compared to the magnitude of error in the baseline. Note, the baseline is the error that exists in the RAP relative to the RAOB. From the RAP and RAOB we begin with a baseline RMSE of 4.432 for the entire profile and a near-surface error of 3.161 (last row of Table 6.1). Thus, a 0.100 difference represents roughly a 3% decrease in error.

Linear and fully-connected models have larger errors compared to the convolutional-based networks. When inspecting the profile estimates from these two models we find the output profiles to be jagged and noisy with measurable changes between every vertical level. This result is due to the relationships of independent variables and the lack of spatial context. However, since spatial relationships are a property of the data, the use of convolutional layers not only produce lower errors but also have smoother profile estimates. **The overall best architecture is the U-Net ar-**

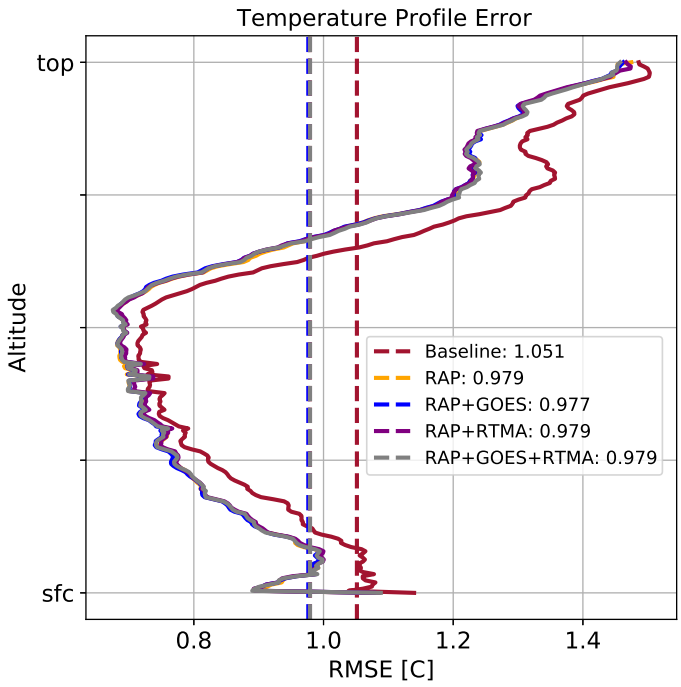
Table 6.1: Error summary for each network and input feature combination using the test dataset. E is the Root-Mean-Squared Error over all output features, and E_{sfc} is only the error of near-surface values. Each grouped column represents the input features used during training and evaluation. The baseline represents the errors between the initial RAP profiles and the RAOBs.

	RAP		RAP+GOES		RAP+RTMA		RAP+GOES+RTMA	
	E	E_{sfc}	E	E_{sfc}	E	E_{sfc}	E	E_{sfc}
Linear	3.605	2.228	3.524	2.235	3.604	2.235	3.519	2.226
FC	3.428	2.194	3.313	2.190	3.428	2.171	3.287	2.173
Conv1D	3.406	2.180	3.291	2.141	3.399	2.151	3.291	2.149
U-Net	3.440	2.120	**3.273**	**2.102**	3.368	2.108	3.300	2.109
Baseline	4.432	3.161						

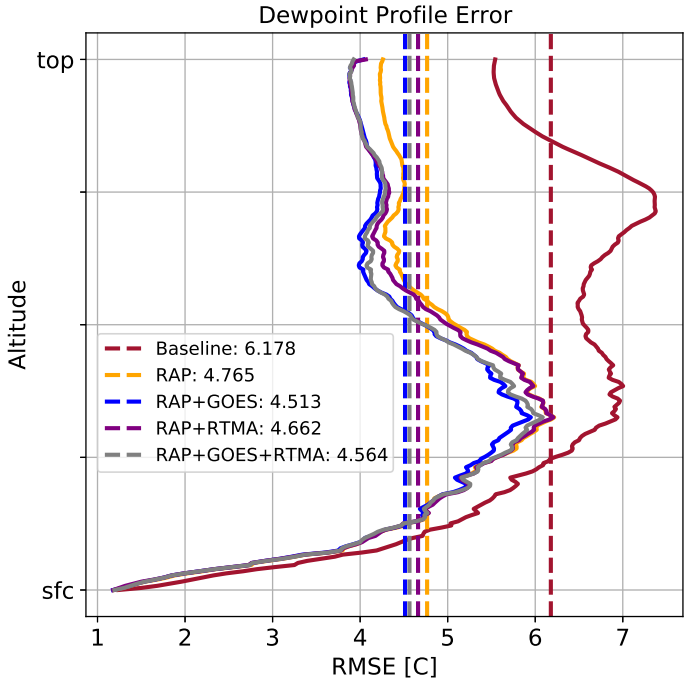
chitecture, which outperforms all other networks when using the RAP and GOES as input data. For the remaining experiments we use this architecture unless otherwise noted.

Figure 6.6 shows the temperature and dewpoint temperature errors across all test samples for each vertical level when varying the input data. The result is four separate models trained with different data with each line revealing where in the profiles the improvements are being made. When comparing to the baseline, the temperature profiles have a relatively steady decrease in error at every vertical level except for near the surface and surrounding $6.5\,\mathrm{km}$ above the surface (Figure 6.6a). Errors directly at the surface have slight improvements followed by a greater decrease until about $2.0\,\mathrm{km}$ above the surface. At about $6.5\,\mathrm{km}$ above the surface, where the RAP has the smallest RMSE, we find the networks to have the least relative improvements. Additionally, across the entire profile there are no significant changes seen when varying the input data. On the contrary, when evaluating the dewpoint profile errors we find the greatest improvements at upper altitudes and with change in the input data (Figure 6.6b). The addition of GOES data as input indicate the greatest improvements between roughly 4–$12\,\mathrm{km}$, whereas the errors are consistent at every other vertical level.

Computing the RMSE over thousands of samples in the test set provides a comprehensive view of the performance of a network, but easily generalizes performance and makes it difficult to isolate particular improvements and limitations. By looking at a specific example we can get another view of the profile estimate characteristics. Figure 6.7 visualizes the Skew-T Log-P from Little Rock, Arkansas (LZK) shortly before 00Z on June 24, 2017 with the RAOB, RAP, and machine learning estimate overlaid. The dewpoint estimate has the most notable improvement, especially in the upper-levels between 400–$100\,\mathrm{mb}$, whereas minor sensitivities in the temperature profile are not captured. In many scenarios the ML estimate is very similar to the initial RAP profile, and in this example, both estimates fail to identify the temperature inversion at $700\,\mathrm{mb}$.



(a) Temperature



(b) Dewpoint

Figure 6.6: Networks trained with different input features showing their errors at every vertical level as compared to the baseline (error between the RAP and RAOB). The vertical dashed lines and respective numbers in the legend represent the RMSE over all vertical levels and samples. (a) Temperature profile errors. (b) Dewpoint profile errors.

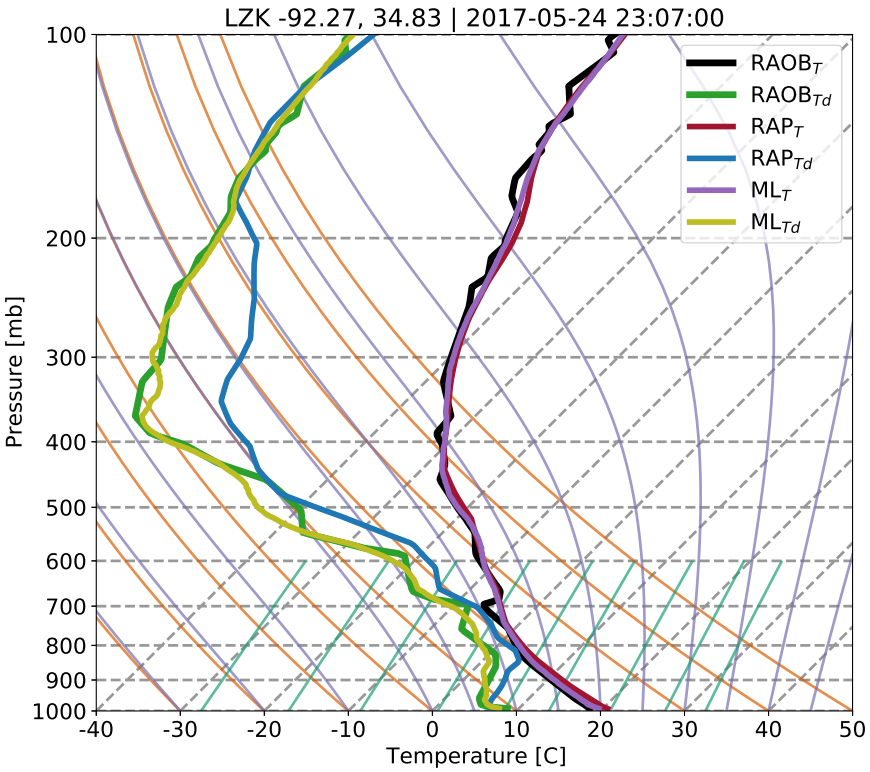


Figure 6.7: A standard Skew-T Log-P diagram of temperature, T, and dewpoint temperature, T_d, profiles for a radiosonde observation (RAOB), collocated profiles from the Rapid Refresh (RAP), and machine learning (ML) estimates over Little Rock, Arkansas (LZK).

Chapter 7

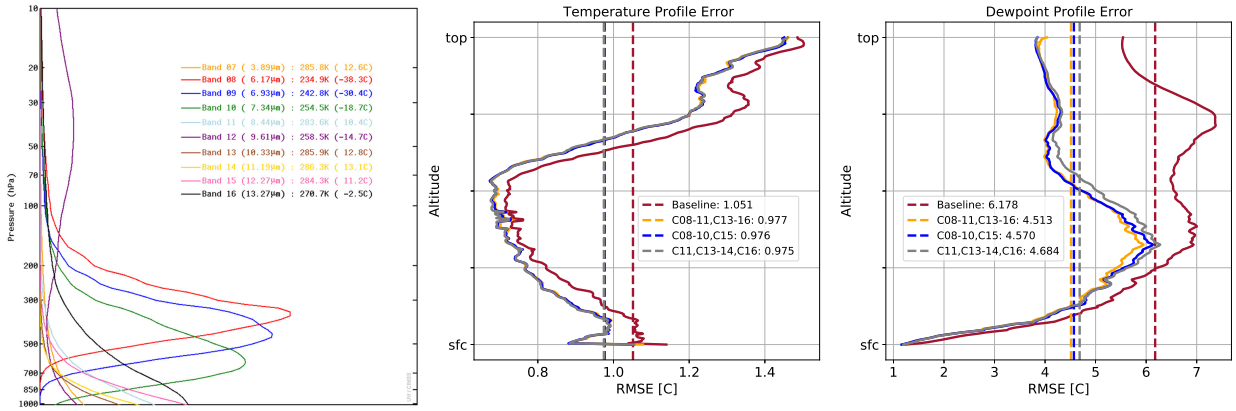
Evaluating Meteorological Conditions

In Chapter 6, the top performing neural network was found to be the Residual U-Net, but we only assessed the general profile accuracy. Here, we use this network architecture to better understand the usage of GOES-16 ABI channels for neural networks, the impact that cloud coverage has on producing accurate profiles, and how different seasons and geographical regions influence performance.

7.1 Importance of GOES-16 ABI

Using the GOES ABI data improves the accuracy of the profile estimates, although it is not initially clear what channels from GOES include the most information and how the network utilizes this data. To provide insight to these challenges we perform an ablation study for different channels in conjunction with closer inspection of the networks' output to see how the networks improve. Foremost, we separate the data by channels into three groups, with (a) all eight channels; (b) only the water vapor bands ($6.2, 6.9,$ and $7.3\,\mu\mathrm{m}$) and near-surface longwave window channel ($12.3\,\mu\mathrm{m}$); and (c) all channels except those in (b), namely $8.4, 10.3, 11.2,$ and $13.3\,\mu\mathrm{m}$. Thereafter, for each group we train a new network initialized with different random weights, all of which contain the same architecture and hyperparameters, and then we evaluate the performance at all vertical levels to contrast how the removal of features influence profile accuracy.

The choice of channels in groups (b) and (c) is selected to understand the relationship between the weighting functions of GOES ABI and what the network learns. The water vapor bands do not see the surface, but they capture information at mid-levels, so we expect the profile errors at these levels to decrease when these channels are used. In Figure 7.1a, the corresponding vertical weighting functions for the U.S. Standard Atmosphere with a satellite zenith angle of 40° is shown. The functions are calculated using a simulated SRF of each wavelength, and they outline the sensitivity of each band to different vertical levels in the atmosphere. For example, with the exclusion



(a) ABI Weighting Functions **(b)** Temperature **(c)** Dewpoint

Figure 7.1: Networks trained with different channel combinations from GOES showing their errors at every vertical level as compared to the baseline (error between the RAP and RAOB). The vertical dashed lines and respective numbers in the legend represent the RMSE over all vertical levels and samples. (a) Vertical ABI IR weighting functions for the U.S. Standard Atmosphere (figure from [50]) (b) Temperature profile errors. (c) Dewpoint profile errors.

of channel 12 ($9.6\,\mu\mathrm{m}$), we can see channel 8 ($6.2\,\mu\mathrm{m}$) peaks the highest at roughly $350\,\mathrm{mb}$ and then falls off at $700\,\mathrm{mb}$. In reality, the weighting functions profile is variable, and may not be as smooth and their peak levels differ slightly between locations or atmospheric conditions. The standard view serves as a reference to understand the typical characteristics of these bands and their importance at different vertical levels.

Our experiments show profile errors that confirm the use of the water vapor bands to aid in improving the mid-levels of the moisture profile. Figure 7.1c illustrates the profile errors at all vertical levels for each network overlaid. Here we see that the network trained only with group (c) performing the worst overall with noticeably higher errors between 4.25–$12.75\,\mathrm{km}$ above the surface. However, when the water vapor bands are added as input to the network (specified by group (b)) there is a clear decrease in profile error at these levels. The region of lower error corresponds to the weighting functions of the provided channels, indicating these channels contain important information that is learned by the networks. Moreover, when using all channels as input, the profiles show the lowest overall error. The same is not seen in the temperature profiles, and adding/removing channels seems not to add noticeable information that make the profiles

more accurate. Overall, including these channels not only demonstrates utility, but also provides appropriate information at the expected levels in the vertical profiles.

7.2 Impact of Cloud Coverage

Samples are separated into clear and cloudy bins as determined by using collocated GOES-R Clear Sky Mask [51]. Cloud masks that contain missing values are flagged and the associated samples are discarded during our analysis. The valid samples are partitioned into respective groups, such that the training set has $15,159$ clear-sky and $11,654$ cloudy-sky samples; validation has $2,046$ clear-sky and $1,517$ cloudy-sky samples; and test has $2,997$ clear-sky and $2,344$ cloudy-sky samples. Effectively, the data has a slight bias toward clear-sky conditions with nearly 57% of the samples constituting the training data. Foremost, we compute a baseline error for the temperature and dewpoint profiles of the RAP under the two conditions. An improvement to the profiles will show errors less than this baseline.

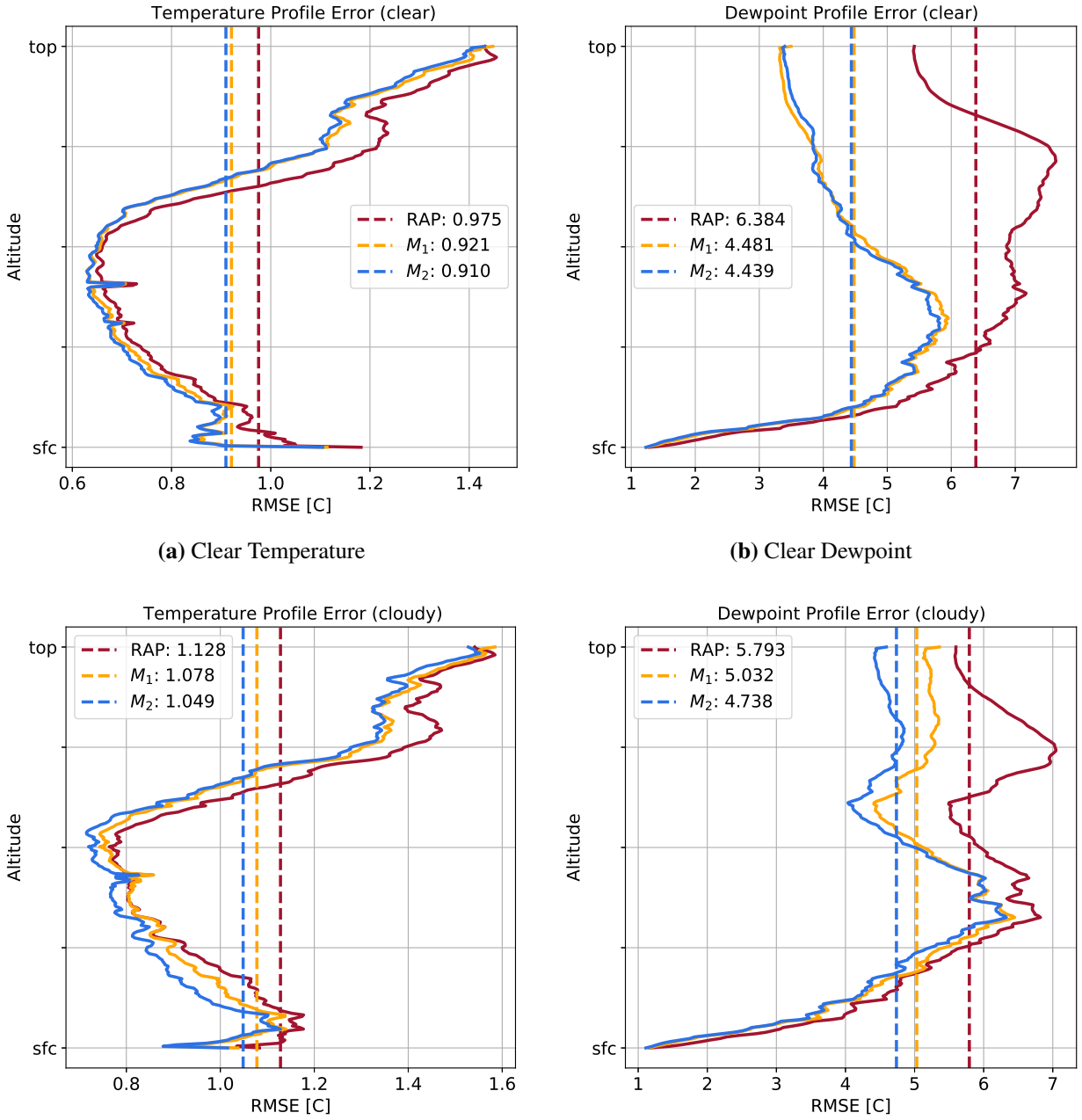
By first observing the baseline we find interesting characteristics regarding where the RAP accuracy degrades. With respect to temperature, the errors directly at the surface are 9.09% lower in cloudy-skies as compared to clear-skies. However, errors in clear-skies are lower at every vertical level above the surface. This result is an artifact of the RAP and its ability to accurately depict the temperature profile in cloudy conditions as initialization variables (*e.g.*, satellite radiances, radar reflectively, etc.) assimilated to the system also lack information in cloudy conditions. On the contrary, dewpoint is more accurate to the RAOBs in cloudy conditions and has a noticeable trend of decreased error between roughly 8–$12\,\mathrm{km}$.

Two models are compared to better understand how cloudy coverage influences profile accuracy and network performance. Both models have an identical U-Net architecture, but the data used for training differs. The first model M_1 takes as input only the RAP and GOES data from clear-sky samples for training, whereas M_2 is the same model from the architecture search that is trained on all available data. Thereafter, we independently evaluate the networks on the test set of

clear and cloudy conditions and report the total RMSE and vertical errors for both profiles. These results are shown in Figure 7.2 with both models compared against the baseline conditions.

Based on the baseline errors in the RAP we can begin to identify potential biases in the dataset. Errors directly at the surface are greater in clear-sky conditions for both profiles, although the near-surface errors in the temperature profiles are greater in cloudy conditions. An initial hypothesis leads us to believe that the disparity of errors in the baseline of the RAP could force M_2 to over(under) correct for surface values when cloudy(clear) conditions are present. However, in Figure 7.2a we find that even M_1 struggles to improve upon M_2 directly at the surface in clear-skies. In the case where the baseline error is low at the surface in cloudy conditions, M_2 only partially outperforms M_1 (Figure 7.2c).

As a general result we find the use of additional data to improve model accuracy and training only on clear-sky samples to provide little to no benefits. Model M_2 contains lower errors for each scenario in our experiments with greater improvements to profiles in cloudy conditions. More specifically, M_1 performs similar to M_2 when tested on clear-sky conditions, but when testing on cloudy-conditions M_1 does not generalize well. In cloudy-conditions, between 4.3–$8.5\,\mathrm{km}$ above the surface, the temperature profile estimates from M_1 are no better than the RAP (Figure 7.2c). Additionally, the dewpoint profile estimates $8.5\,\mathrm{km}$ above the surface have errors roughly 0.5 °C higher than M_2 (Figure 7.2d). Overall, using all the available data during training creates a model that generalizes better to various atmospheric conditions and outperforms a model trained only on clear-sky conditions.



(a) Clear Temperature

(b) Clear Dewpoint

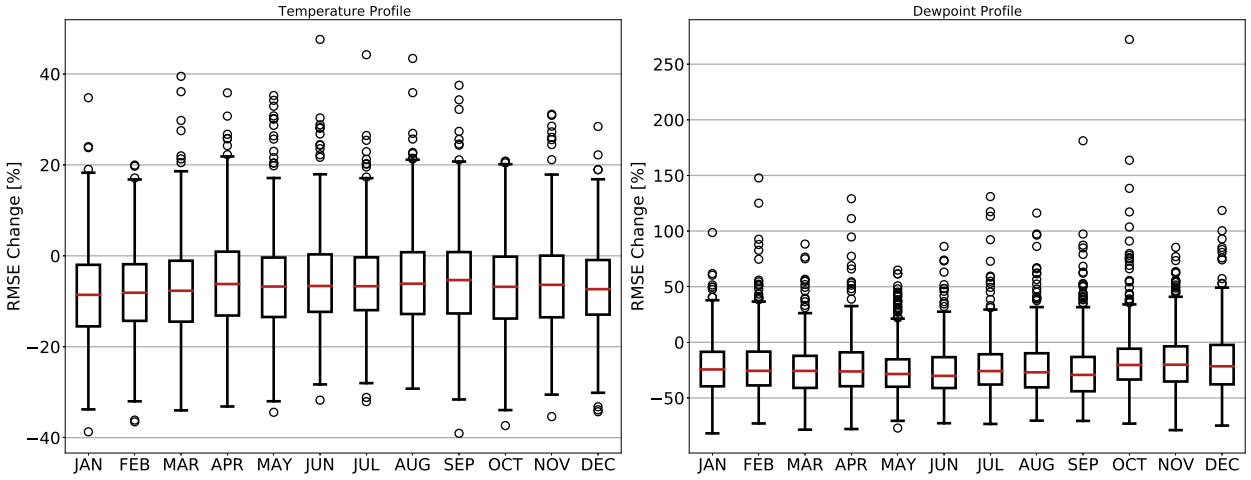
(c) Cloudy Temperature

(d) Cloudy Dewpoint

Figure 7.2: Networks trained and evaluated on different cloud conditions showing their errors at every vertical level as compared to the baseline (denoted RAP, which is the error between the RAP and RAOB). M_1 takes as input only the RAP and GOES data from clear-sky samples for training, whereas M_2 is trained on all available data. The vertical dashed lines and respective numbers in the legend represent the RMSE over all vertical levels and samples. (a) Clear temperature profile errors. (b) Clear dewpoint profile errors. (c) Cloudy temperature profile errors. (d) Cloudy dewpoint profile errors.

7.3 Seasonal Influence

As another way to evaluate the performance of our model, we look to see if the model biases any particular season during training. The data are partitioned temporally such that no one dataset contains a disproportionate number of observations for a given month, and only the test data are considered for evaluation. There exists a natural bias in the number of observations for the months of September through December as samples are not yet collected for the latter half of the year in 2020. However, there are more observations in North American spring and autumn months with April and May having the highest counts. The increase of observations during these months are due to the interest of peak severe weather seasons, which takes place in March, April, and May.



(a) Temperature **(b)** Dewpoint

Figure 7.3: Each sample represents the percent decrease in error from the ML estimate error relative to the baseline error. A lower value shows a better improvement to the profiles. (a) Temperature profile errors. (b) Dewpoint profile errors.

Initial baseline profile errors within the test set show the greatest errors in the RAP for the temperature profiles in December through April and for the dewpoint profiles in the months of April through September (not shown here). Figure 7.3 shows a boxplot for the percent change in RMSE of the machine learning estimate with the RAOB relative to the baseline RMSE of every sample. The median and spread of values are fairly consistent for every month, even though the

RAP profiles have higher errors in certain months. As a result, the network is not learning to bias a particular month or focus on learning particular patterns for only low- or high-severity weather seasons.

For each boxplot, the whiskers mark the bounds for statistical outliers, and are found with a standard distance of $1.5 * (IQR)$ units below the first quartile, Q_1, and above the third quartile, Q_3. Based on these quartiles we can see a general trend where 50% of the variability around the median is less than zero, and nearly 25% of values (greater than Q_3) have profile errors that are larger than the baseline. Interestingly, the dewpoint profiles have a larger range over all of the months with greater decreases/increases in error.

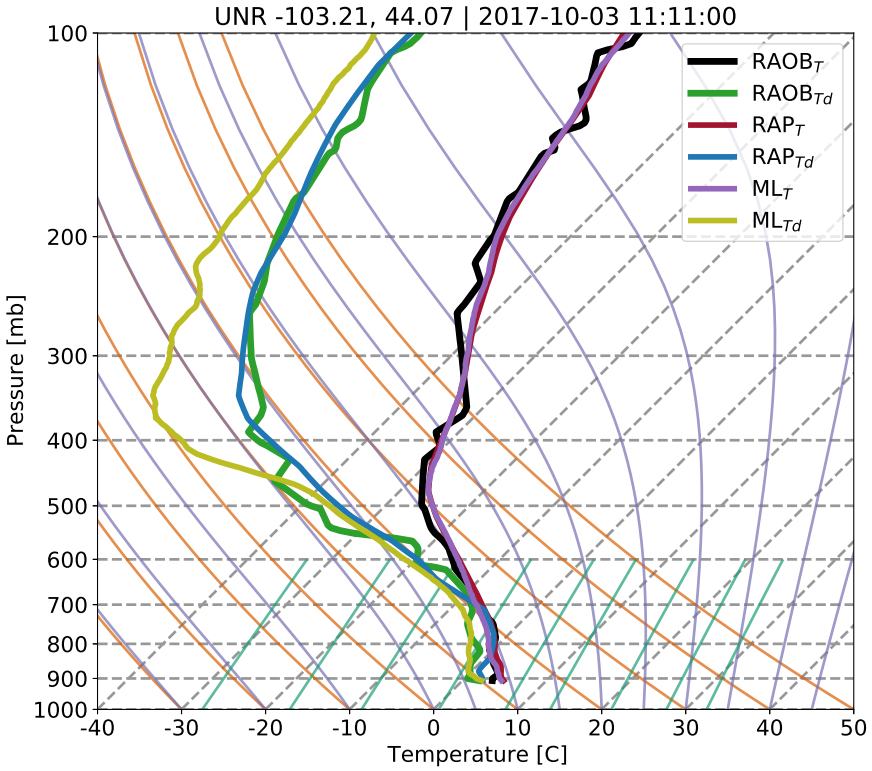


Figure 7.4: An under performing example of an ML estimate for the temperature, T, and dewpoint temperature, T_d, profiles over Rapid City, South Dakota (UNR).

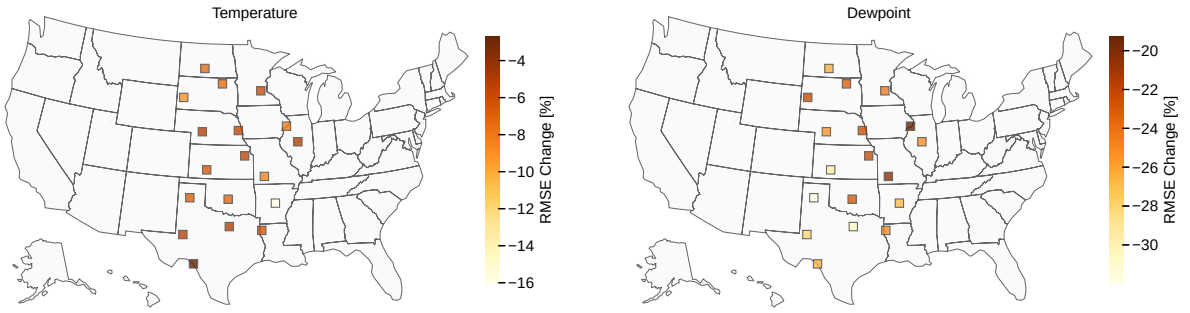
The most significant outlier seen in Figure 7.3b shows a particular profile with over 250% higher error than the original RAP profile. This radiosonde was launched from Rapid City, South Dakota (UNR) shortly before 12Z on October 3, 2017. Figure 7.4 is a Skew-T diagram with the

overlaid RAOB, RAP, and machine learning estimate for the temperature and dewpoint profiles. In this example, the RAP profile is an already semi-accurate initial guess for the vertical profile of both variables. However, the network's output significantly underestimates the upper-levels of the dewpoint profile. This is likely due to the lack of generalizability of the network and the dataset containing few samples where the upper-levels of the dewpoint profile in the RAP is actually accurate with the RAOB to begin with.

7.4 Regional Performance

Of the 18 locations used in this study only Del Rio, Texas (DRT) has fewer than $2,000$ samples (totalling $1,421$), and conversely, the greatest number of samples is from Little Rock, Arkansas (LZK) totalling $2,271$ samples. All other locations are well balanced in the number of samples around the mean of 2130.833. By inspecting the performance of the network at each individual location in the test set we are looking to see if a particular location is biased during training. As previously mentioned, and prior to training, the data are partitioned to account for the distribution of samples at each location so we remove any prior sample imbalances. As such, each dataset partition maintains a proportionate distribution of samples with DRT having the fewest observations and LZK with the greatest.

From the RAP data we first compute the initial baseline RMSE with the RAOBs for every location to understand where the greatest/lowest errors are. In the temperature profiles of the RAP, the northwest region of observed locations has the lowest errors, whereas the western most locations between South Dakota and Texas have the greatest errors. In particular, Rapid City, South Dakota (UNR) and DRT have the largest RMSE of 1.234 and 1.195, respectively. For the dewpoint profiles, there exists a regional pattern where errors increase in southern locations with the northern region having the lowest errors (now shown here). To measure the improvements for these locations we compute the percent decrease RMSE between the baseline and machine learning estimate errors. If the network is biased toward any one location then the locations with the greatest errors or largest number of samples will have the lowest errors and large percent change.



(a) Temperature **(b)** Dewpoint

Figure 7.5: Each location shows the percent decrease in error from the ML estimate error compared to the baseline error. A lower value shows a better improvement to the profiles. (a) Temperature profile errors. (b) Dewpoint profile errors.

Figure 7.5 illustrates the change in error for the temperature and dewpoint profiles individually across all locations. No visual patterns or regions are seen with a general bias of profile improvements, albeit there are a few locations with significantly higher/lower change in errors. In Figure 7.5a DRT has the lowest percent decrease in RMSE, but it was also the location with one of the higher baseline errors and fewest number of samples in the dataset. LZK has the greatest improvement to the temperature profile with moderate initial error and it is also the location with the most data samples. As a result, we speculate that the network benefits more when a location has a greater number of samples, whereas the locations with a higher initial error are not necessarily prioritized. The same observation is not consistent with the dewpoint profiles. In fact, Amarillo, Texas (AMA) has the largest initial error and the most pronounced improvements to the profiles, but a near average number of samples. Furthermore, in Figure 7.5b, we see the locations with the highest initial error for dewpoint (southwest region) to have the greatest improvements overall. Lastly, the dewpoint profiles have a larger magnitude of change at every location as compared to that temperature profile, which is an observation discussed previously. Overall, in some scenarios, the network appears to favor the number of samples, whereas in other situations, the locations with the highest baseline errors see the greatest improvements.

Chapter 8

Modeling Sounding Products Directly

Meteorologists often use Skew-T diagrams to derive atmospheric indices, which prove to be useful during inclement weather and severe weather situations. These indices change rapidly over the course of a couple of hours due to the change of thermodynamics in the atmosphere. Therefore, having an abundance of NWP profiles allows for more consistent interpretations over the twice-daily radiosonde observations. However, as previously discussed, accuracy in the NWP profiles are critical to generating accurate derived indices, but estimating more accurate indices directly is also desirable to meteorologists. In this chapter we explore the use of neural networks to map NWP profiles to ground truth indices of the RAOBs and skip the intermediate step of correcting the entire profile.

8.1 Convective Products

Many products or indices can be derived from the Skew-T diagrams to guide meteorologists in forecasting convective weather. In meteorology, the measures of convective available potential energy (CAPE) and convective inhibition (CIN) are standard indicators of the potential of convective instability found from the temperature and moisture profiles. Large CAPE values indicate high vertical velocities in the updraft region of a thunderstorm and reflect positive buoyancy. The value increases with daytime heating, near-surface advection of warm air, cooling temperatures in the mid-levels, and increased near-surface moisture. Large CIN values represent atmospheric stability and are decreased (further from zero) with a large cap strength or a dry planetary boundary layer.

CAPE is computed as the area between the parcel temperature, T_{parcel}, and the cooler environmental temperature, T_{env}, bounded by the level of free convection (LFC) and the equilibrium level (EL). Using a formula adopted from [49], we mathematically calculate this as:

$$CAPE = -R_d \int_{LFC}^{EL} (T_{parcel} - T_{env}) dln(p), \tag{8.1}$$

where $R_d = 287.058\,\mathrm{J\,kg^{-1}K^{-1}}$ is the dry air gas constant and p is atmospheric pressure from the sounding. Thus, prior calculations of the air parcel, LFC, and EL are required before computing CAPE. The LFC is found with the first intersection for the path of an ideal air parcel and the measured environmental temperature. If the LFC is found to be below the lifting condensation level (LCL), then the parcel must first ascend dry adiabatically until saturation before rising further to the LFC. The EL is the last intersection of the air parcel and the environmental temperature.

CIN measures the energy needed to lift a parcel of air from the surface to the LFC. Using the LFC calculated prior, we find the area between the ideal temperature parcel and a warmer environmental temperature by integrating between the surface and the LFC:

$$CIN = -R_d \int_{SFC}^{LFC} (T_{parcel} - T_{env}) dln(p). \tag{8.2}$$

Figure 8.1 illustrates a Skew-T Log P diagram for a RAOB from Rapid City, South Dakota (UNR) shortly before 00Z on September 4, 2019 with CAPE and CIN values of 2149.867 and $-140.830\,\mathrm{J\,kg^{-1}}$, respectively. In this particular example, the environment around UNR has moderate instability and large inhibition, according to thresholds from NOAA's Storm Prediction Center. In the RAP sounding the CAPE and CIN values are found to be 1942.105 and $-50.001\,\mathrm{J\,kg^{-1}}$, respectively. Both of these values are smaller than the ground truth RAOB. Thus, demonstrating an example for how slight inaccuracies in the RAP can influence accuracy of derived indices.

We generate an auxiliary dataset comprising the CAPE and CIN values for every sounding in the RAOB and RAP profiles. As an additional preprocessing step, the samples containing erroneous derived indices from the RAOB are removed, as defined by: CAPE > $7000\,\mathrm{J\,kg^{-1}}$, CAPE < $0\,\mathrm{J\,kg^{-1}}$, and CIN > $1000\,\mathrm{J\,kg^{-1}}$. A total of 883 samples are removed from the dataset after filtering. Interestingly, the distribution of CAPE and CIN values are non-normal and are strongly skewed with the majority of values centered around zero. Note that CAPE is greater than zero and is positively skewed, whereas CIN has a negative skew.

Summary statistics of values for CAPE and CIN of the RAOBs are reported in Table 8.1 to emphasize the severity of zero value indices and statistical outliers. Here we see that the mean is

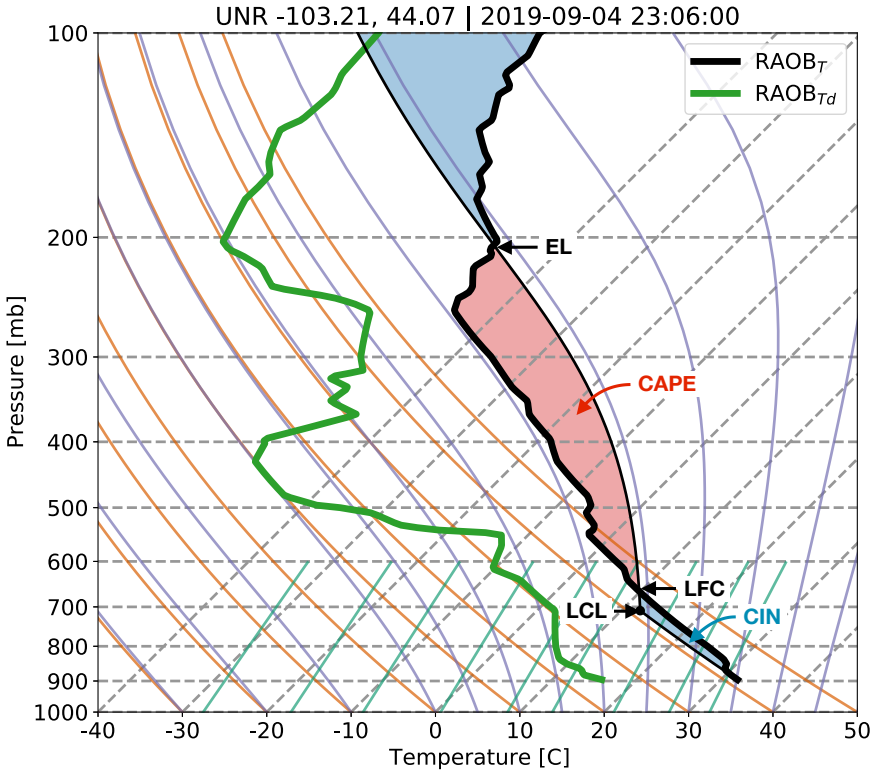


Figure 8.1: A standard Skew-T Log-P diagram of temperature, T, and dewpoint temperature, T_d, profiles with shaded regions of CAPE $= 2149.867\,\mathrm{J\,kg^{-1}}$ and CIN $= -140.829\,\mathrm{J\,kg^{-1}}$ for a radiosonde observation over Rapid City, South Dakota (UNR). Note, CIN is only the blue shaded area under the LFC, and the blue shaded region above the EL is only an artifact of plotting.

significantly different than the median value (which is zero for both variables). Additionally, since CIN is strictly negative, the first quartile represents only 25% of values less than $-1.561\,\mathrm{J\,kg^{-1}}$. Similarly, with values for CAPE only 25% of the data have values found to be greater than $117.115\,\mathrm{J\,kg^{-1}}$. **The distribution of data proves to be a challenge for a neural network to accurately model since the majority of high values of CAPE/CIN are statistical outliers.**

Table 8.1: Summary statistics of CAPE and CIN values of the RAOBs. The derived indices are highly skewed with mainly zero value indices.

	Mean	SD	Min	Q1	Median	Q3	Max
CAPE	332.917	780.470	0	0	0	117.115	6940.407
CIN	-47.432	130.384	-999.684	-1.561	0	0	0

8.2 Model Setup and Data Usage

A convolutional neural network is trained using z-score normalized 1-dimensional profiles of pressure, temperature, and dewpoint temperature of the RAP to estimate the CAPE and CIN values associated with the collocated RAOB. This network structure has six 1-dimensional convolutional layers of size $[32, 32, 32, 64, 64, 64]$, each followed by a 1-dimensional max pooling layer to reduce the profile size by two, and the output of the last layer is input to a fully-connected hidden layer with size $[128]$, followed by a linear output layer of two units. After every convolutional and fully-connected layer is the ReLU activation function to introduce non-linearity. For 45 epochs, and using a batch size of 64, the network optimizes the Huber loss defined piecewise by,

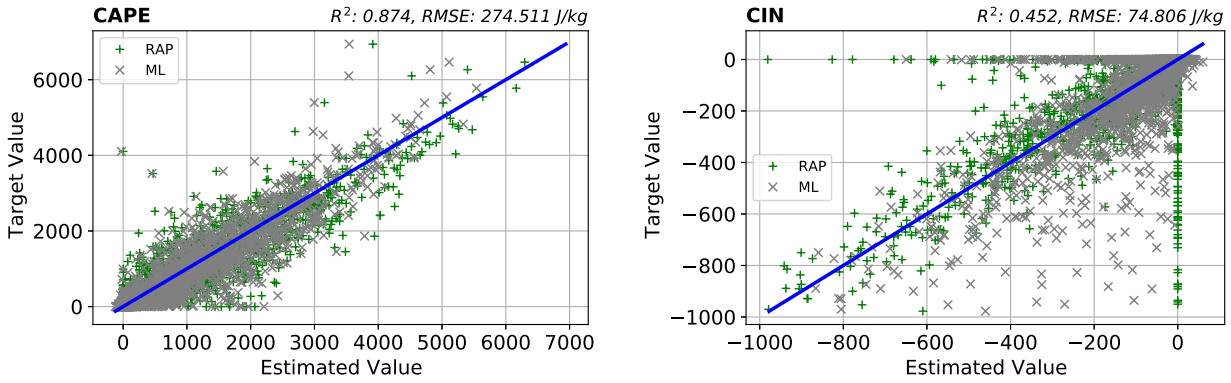
$$\begin{aligned} \mathcal{L}_\delta(a) &= \begin{cases} \frac{1}{2}(a)^2 & \text{for } |a| \leq \delta, \\ \delta\,|a| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases} \\ \mathcal{L}_{\text{Huber}} &= \frac{1}{n}\frac{1}{k}\sum_{i=1}^{n}\sum_{j=1}^{k}\mathcal{L}_\delta(t_{ij} - y_{ij}), \end{aligned} \tag{8.3}$$

where $\delta = 1.0$, using the Adam optimizer and a learning rate of 0.0001. The Huber loss is quadratic for $|a| \leq \delta$, which is equivalent to the MSE, linear for larger values of a, and has equal cases when $|a| = \delta$. This function is less sensitive to outliers in data than MSE since less weight is assigned to large residuals. Through experimentation, the Huber loss outperforms networks trained using MSE and MAE when estimating the sounding products directly.

8.3 Neural Network Performance

From the RAP derived indices we compute baseline metrics with the test data to better understand the inaccuracies in the RAP profiles. The R^2 and RMSE share a summary of how similar the derived indices are with the RAOB, but they are not comprehensive enough in explaining what the errors are. Therefore, we also visualize the target versus estimated values (from the neural network and RAP) to show how similar the results are to the RAOB indices. Comparing to the target values in of the RAOB, the RAP primarily overestimates CAPE and underestimates CIN. This observa-

tion is seen in Figure 8.2, with the most severe cases found when CAPE and CIN both have a value of $0\,\mathrm{J\,kg^{-1}}$. In these figures, a perfect model will show the estimates on the one-to-one line, which directly corresponds to the target values of the RAOB. In Figure 8.2b the CIN values of the RAP show the most extreme errors with values of nearly $-950\,\mathrm{J\,kg^{-1}}$ when the RAOB shows no CIN. Baseline errors in CAPE are evident as well, but seen on a larger scale. More specifically, Figure 8.2a does not show the same large visual errors as with CIN, but with the larger range of values in CAPE, errors as large as $4000\,\mathrm{J\,kg^{-1}}$ are seen in the RAP.



(a) CAPE **(b)** CIN

Figure 8.2: Target RAOB indices versus estimated indices from the RAP and ML estimates. A one-to-one line (in solid blue) represents when the estimated values are identical to the target indices. R^2 and RMSE metrics are reported for the ML estimates in the top right of each figure. (a) Estimated CAPE, (b) Estimated CIN.

In general, the trained network corrects for these over- and under-estimations and demonstrates improvement over the derived indices of the RAP. In Figure 8.2b, when the RAP estimates a CIN of $0\,\mathrm{J\,kg^{-1}}$, the network has an improvement toward the target values. However, the network is unable to bring these values all the way to the observed values. In a few scenarios when significantly large values are present, the network still under-estimates the target RAOB indices. With the estimated values of CAPE, the ML estimates are closer to the one-to-one line when compared to the RAP, but the outliers are not significantly different. In fact, with large CAPE values in the RAOB when the RAP underestimates, the ML estimates are no different than the RAP. Furthermore, with large

CAPE values in both the RAP and RAOB, the ML estimates are lower than the RAP, bringing the difference in values further from the ground truth. Lastly, in these figures we see the ground truth values never have negative CAPE or positive CIN. However, in Figure 8.2a the ML estimates spill into invalid values of CAPE (as seen in the lower left corner), and the same is true for the CIN being greater than zero for some values (upper right corner). This is likely a result of the model slightly overfitting on the training data and failing to accurately learn the constraints of valid indices, thus, estimating values for the test data, which have never been seen before, that are outside the valid range.

Overall performance metrics are shown in Table 8.2 for the baseline RAP indices, ML estimates, and the derived indices from the estimated profiles. The machine learning profiles are those produced in Chapter 5 using the best network architecture. We compute CAPE and CIN values for the profiles and compare the values with both the baseline and ML estimates. In addition to computing R^2 and RMSE for all values in the test data, we also compute the same metrics for a split of the test data where the derived indices in both the RAP and RAOB are strictly greater than zero. The reason for assessing non-zero values is to get a picture of how "more important" data samples are treated. Additionally, we want to ensure the network is not only learning the zeros in the data, but is actually learning large CAPE and CIN values from the profiles.

The comparison of the baseline with the indices computed from the machine learning profiles show similar R^2 for CAPE with slightly lower RMSE, but metrics for CIN with worse measurements for all test data and non-zero values. Thus, inaccuracies at or near the surface up through the LFC are still evident. However, we see more promising results for CIN by directly estimating the derived indices. R^2 and RMSE values are lower than the baseline RMSE for both output variables, indicating the network explains slightly less variance than RAP but more than the machine learning profiles for CAPE. As is, the indices from ML profiles improve CAPE but not CIN; however, by directly estimating these products we see improvements to both indices. Overall, the machine learning based approaches demonstrate the ability to learn these parameters, and by directly estimating them, we can improve upon CIN and get near the CAPE baseline.

Table 8.2: Statistics for CAPE and CIN values of derived indices from machine learning estimates compared to the values of RAOB indices over the test data. Indices of the baseline RAP profiles are found using the original RAP data, ML profiles represent the indices from the corrected profile estimates, and the direct ML estimates are found by directly estimating the indices. Column subscripts with a z denote statistics for the test data where RAP and RAOB indices are greater than zero.

	R^2	RMSE	R^2_z	$RMSE_z$
CAPE				
From Baseline RAP Profiles	0.876	282.068	0.834	437.772
From ML Profiles	0.874	265.297	0.835	411.316
Direct ML Estimates	0.874	274.511	0.826	422.073
CIN				
From Baseline RAP Profiles	0.531	82.805	0.784	92.009
From ML Profiles	0.409	85.672	0.630	111.898
Direct ML Estimates	0.452	74.806	0.586	103.812

Chapter 9

Conclusion

The work in this thesis is pertinent to the emerging research in machine learning for atmospheric science and also meteorologists concerned with forecasting near-term convective threats. As such, we discuss an overview of the results, some guidelines and limitations, and possibilities for future work.

9.1 Discussion

The neural network architectures and evaluations presented in this work share several significant contributions as the first known work of using neural networks to directly improve vertical profiles from an NWP model. By exploring various architectures; linear, fully-connected, convolutional, and a Residual U-Net networks, we outline methods to incorporate signal-like 1-dimensional profiles with 2-dimensional image data in a unified network for signal-to-signal processing and profile enhancements. Through an extensive network search we demonstrate the use of the U-Net to outperform other network architectures by incorporating observational data with encoded profile features in the bottleneck of the network and using a residual connection with the input and output layers. This solution to joining multiple datasets produces more stable training, a model containing fewer parameters, and lower errors than the other multi-input architectures discussed within.

A second contribution made in this work is detailing how domain knowledge can be incorporated into training neural networks. This is done by utilizing observational data features that are important indicators of atmospheric conditions, designing a network structure that best suits these features, and exploring loss functions that emphasize important characteristics of vertical profiles. Results show the near-surface weighting loss to perform best, but we also outline how derived indices, such as total precipitable water, can be included as an additional term in the loss function. Additionally, we identify the important features in observational data from surface measurements

and satellite data that can be learned by neural networks. While the usage of GOES-16 ABI is well understood in atmospheric science, this work explicitly shows how machine learning leverages the water vapor bands ($6.2, 6.9,$ and $7.3\,\mu\mathrm{m}$) to reduce profile errors and increase accuracy of the moisture profiles at mid-level altitudes. Additionally, the experimental data shows the use of RTMA to have little to no benefit when training a model. This result is also significant as the surface data was found to contain high variability when compared to ground truth RAOBs, which effectively offers no additional information when used as input with neural networks.

Using the proposed U-Net architecture, we assess the impact of cloud coverage, the effect of seasonality of the data, and how geographical location may alter results. By evaluating multiple models, we find that using both clear- and cloudy-sky samples during training improves model performance under both conditions. Additionally, this model outperforms one trained only with clear-sky samples, which shows the use of additional data to be helpful and the model trained on clear-sky samples does not perform well if cloudy. By looking at errors during different months of the year, the samples are treated equally with a similar decrease of error, irrespective of months that contain samples with higher initial errors, indicating no bias of seasonality. Lastly, the experimental data verifies that locations that contain more samples show greater percentage improvements with the temperature profiles and locations with higher baseline errors have the greatest percentage improvements with the dewpoint profiles.

Overall, the use of neural networks for improving vertical profiles of the RAP successfully produces profiles that are more similar to RAOBs with a consistent decrease of error in the temperature profiles and more outstanding improvements in mid- to upper-level measurements of the dewpoint temperature profiles. The associated derived indices result in more accurate CAPE values, but slightly worse CIN values, which indicates measurements at and near the surface have room to be improved. Additionally, by directly estimating CAPE and CIN, we can improve upon CIN over the ML profiles, but estimates of CAPE are not necessarily better. With stable improvements among the profiles, especially in the moisture profile, we provide a step toward increasing the reliability of accurate NWP profiles. Furthermore, the architecture and methods we present to

improve profiles will potentially aid forecasters in conducting more accurate near-term convective threat assessments.

9.2 Limitations

Generalizability is a significant limitation with training a performant neural network in this work. In our experiments we find that simpler architectures, by means of fewer parameters and lower computational complexity, are unable to capture the patterns in the vertical profiles and are generally an inadequate choice. Conversely, models are more prone to overfit on the training data as the complexity of the architecture increases. While there are methods to reduce overfitting, we find there to be a middle ground where slightly reducing the complexity of the model outperforms more complex networks trained to reduce overfitting. However, we are unable to correct for all the errors in the RAP at different vertical levels and unseen data samples.

With the best performing U-Net architecture the RAP has a general improvement for both the temperature and dewpoint profiles. However, this improvement is not uniformly seen across all samples, and the network does a better job at correcting the larger errors in the dewpoint profile. When observing the change of error in the profiles, as was done in Section 7.3, there exists nearly 25% of the data that has a decline in accuracy as compared to the accuracy of the baseline RAP errors. This result is likely a mixture of the networks inability to generalize to all of the data and the data containing a significant amount of noise and variability. Additionally, in some scenarios where there exist fewer samples for a given radiosonde location there exists larger errors in the profiles. We speculate that the use of more data will diminish sample imbalance, improve generalization, and allow for more complex networks to achieve better results.

9.3 Technical Challenges

This thesis overcomes many technical challenges that are noteworthy to guide the continuation of related research. A significant portion of this work alludes to the performance and practicability of neural networks for our particular application. However, a great deal of time was devoted to or-

ganization and usability of data throughout the process. The datasets comprise $9.79\,\mathrm{TB}$ of GOES ABI, $2.10\,\mathrm{TB}$ for the RTMA, $734\,\mathrm{GB}$ for the RAP, and $2.5\,\mathrm{GB}$ for the RAOBs. Foremost, the acquisition of the RAP data was particularly difficult. Raw model data are stored, without chronological ordering, on tape drives on NOAA's High Performance Storage System (HPSS), which we transferred to NOAA's high performance computer, HERA. Files on HERA were uncompressed and converted to a CONUS Lambert Conformal grid, and then transferred to the storage device at the Cooperative Institute for Research in the Atmosphere (CIRA) for our use. Initial transfer times to get data from HPSS to CIRA took 24 hours to extract 14 days of RAP data. We were able to achieve a $2\times$ speedup of end-to-end transfers by first locating all the tape locations on HPSS for a given date range, and then extracting all the dates within the range on one tape drive at a time. Nevertheless, this process of acquiring the RAP took nearly four months to transfer data for 2017-2020.

Furthermore, while disk input/output is traditionally the bottleneck in computing, we find the computational overhead of extracting fields from the map projections to be the most expensive component during data preprocessing. To improve preprocessing times, we employ caching and concurrent processing practices to maximize the efficiency of gathering only relevant information. Each RAOB need only to be read from disk once, but since multiple radiosondes are launched surrounding a particular date and time for different locations, we can load the additional datasets to memory more effectively so they too are read only once. The GOES, RTMA, and RAP datasets are on a spatial grid and so they can be cached to memory while time sorted RAOBs are read sequentially. Furthermore, since each sounding is temporally separable, we organize data concurrently on separate threads over different months to further reduce computational time.

Lastly, within the aforementioned experiments, we describe a total of 1,320 trained neural networks from the linear to more complex convolutional architectures. Not mentioned are hundreds of other models designed with different normalization techniques, hyperparameters, network architectures, and model feature inputs. Within this work, only the top performing configurations are reported, albeit other unexplored techniques may exist with superior results (*e.g.*, generative

adversarial networks, recurrent neural networks, etc), although we believe experimenting with simpler network architectures and building upon them is critical to understand the limitations and advantages of a given network. In general, the significance of deeply exploring the search space is two-fold, that is (a) to provide a robust result of several weight initializations and (b) improve confidence in a particular model design with justification on prior experimentation. However, such exploration requires an abundance of computational resources and time for models to train.

9.4 Possible Future Work

As discussed in Chapter 8, there are additional properties and derived indices that can be computed from the vertical profiles. These indices describe the conditions and measurements of the environment and their accuracy directly corresponds to the accuracy of the profiles. With CAPE and CIN, many levels in the profile directly contribute to its calculation (*e.g.*, temperature and dewpoint surface values and the LCL/LFC/EL). This information which describes the profile could be useful in incorporating the accuracy of the indices into the loss function for the neural network to optimize, similar to how we compute TPW in the loss described in Section 5.2. Here, we will also maintain errors over the profile with use of MSE or MAE as it accounts for the residuals of every output feature, and then combine one of these losses with the minimization of derived indices such as CAPE and CIN. We mathematically represent this function as:

$$\begin{aligned} \mathcal{L}_{\text{CAPE\_CIN}} &= \frac{1}{n}\sum_{i=1}^{n}(\text{CAPE}_{t_i} - \text{CAPE}_{y_i})^2 + (\text{CIN}_{t_i} - \text{CIN}_{y_i})^2, \\ \mathcal{L}_{\text{CMSE}} &= \mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{CAPE\_CIN}}, \end{aligned} \tag{9.1}$$

where CAPE and CIN are computed for the target RAOB, t_i, and estimate, y_i for each sample $i = 1, \ldots, n$. The disadvantage to this method is the calculation for CAPE and CIN is computationally expensive with nearly $300\,\text{ms}$ of additional overhead per profile (from our experiments). In turn, this would yield hours to track history of each epoch and potentially days to train the model.

Alternatively, a derived product such as the K-index and Total Totals Index could substitute for CAPE/CIN, but neither include surface information, and may not be as effective.

Recall that with certain profiles, the errors increased relative to the errors in the RAP. This characteristic of model performance is of interest to understand. One possible method would be to identify any reoccurring properties where errors are high in the baseline errors of the RAP. More so, it would be beneficial to locate fine grain patterns where errors are high and atmospheric properties are not captured. For example, the RAP does not accurately estimate the cap and temperature inversions in the profiles, and it would be helpful to understand if the magnitude of these inversions are accurately captured by the profile estimates.

Lastly, in many machine learning applications the idea of "transferring knowledge" via transfer learning is employed to improve model performance. This approach has been shown to be effective when the target training set has few samples or has high dimensionality. As such, in an effort to improve vertical profiles, a preliminary model could be trained on augmented RAOB samples from historical launches between the years of 1980 to 2017. The target observation should remain unaltered, but input samples could be smoothed using linear interpolation and value shifts to be more like NWP output. Thereafter, the last layer of the network is removed and replaced with an output layer of randomly initialized weights. The model is then trained again using the NWP model as input with data from the years of 2017 to 2020. Effectively, the pretrained network's parameters have a general idea of what the radiosonde profiles look like, and by fine tuning with the NWP model, we can potentially increase generalizability and accuracy of the profiles.

Bibliography

[1] Melissa M. Hurlbut and Ariel E. Cohen. Environments of northeast u.s. severe thunderstorm events from 1999 to 2009. *Weather and Forecasting*, 29(1):3–22, February 2014.

[2] P. J. Kocin, L. W. Uccellini, and R. A. Petersen. Rapid evolution of a jet streak circulation in a pre-convective environment. *Meteorology and Atmospheric Physics*, 35(3):103–138, 1986.

[3] James T Moore. The forcing and evolution of the three dimensional moisture convergence during the 10–11 april 1979 severe weather outbreak. In *Preprints, 12th Conf. on Severe Local Storms, San Antonio, TX, Amer. Meteor. Soc*, pages 209–212, 1982.

[4] Ariel E Cohen, Steven M Cavallo, Michael C Coniglio, Harold E Brooks, and Israel L Jirak. Evaluation of multiple planetary boundary layer parameterization schemes in southeast us cold season severe thunderstorm environments. *Weather and Forecasting*, 32(5):1857–1884, 2017.

[5] Ruiyu Sun, Steven K Krueger, Mary Ann Jenkins, Michael A Zulauf, and Joseph J Charney. The importance of fire–atmosphere coupling and boundary-layer turbulence to wildfire spread. *International Journal of Wildland Fire*, 18(1):50–60, 2009.

[6] Ismail Gultepe, R. Sharman, Paul D. Williams, Binbin Zhou, G. Ellrod, P. Minnis, S. Trier, S. Griffin, Seong. S. Yum, B. Gharabaghi, W. Feltz, M. Temimi, Zhaoxia Pu, L. N. Storer, P. Kneringer, M. J. Weston, Hui ya Chuang, L. Thobois, A. P. Dimri, S. J. Dietz, Gutemberg B. França, M. V. Almeida, and F. L. Albquerque Neto. A review of high impact weather for aviation meteorology. *Pure and Applied Geophysics*, 176(5):1869–1921, May 2019.

[7] Benjamin F. Zaitchik, Jason Evans, and Ronald B. Smith. MODIS-derived boundary conditions for a mesoscale climate model: Application to irrigated agriculture in the euphrates basin. *Monthly Weather Review*, 133(6):1727–1743, June 2005.

[8] Joseph B Olson, Jaymes S Kenyon, Irina Djalalova, Laura Bianco, David D Turner, Yelena Pichugina, Aditya Choukulkar, Michael D Toy, John M Brown, Wayne M Angevine, et al. Improving wind energy forecasting through numerical weather prediction model development. *Bulletin of the American Meteorological Society*, 100(11):2201–2220, 2019.

[9] Nils Gustafsson, Tijana Janjić, Christoph Schraff, Daniel Leuenberger, Martin Weissmann, Hendrik Reich, Pierre Brousseau, Thibaut Montmerle, Eric Wattrelot, Antonín Bučánek, et al. Survey of data assimilation methods for convective-scale numerical weather prediction at operational centres. *Quarterly Journal of the Royal Meteorological Society*, 144(713):1218–1256, 2018.

[10] Timothy J Schmit, Jun Li, Su Jeong Lee, Zhenglong Li, Richard Dworak, Yong-Keun Lee, Michael Bowlan, Jordan Gerth, Graeme D Martin, William Straka, et al. Legacy atmospheric profiles and derived products from goes-16: Validation and applications. *Earth and Space Science*, 6(9):1730–1748, 2019.

[11] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[12] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. John Wiley & Sons, 1949.

[13] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[14] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[15] Rohit Chakraborty and Animesh Maitra. Retrieval of atmospheric properties with radiometric measurements using neural network. *Atmospheric Research*, 181:124–132, 2016.

[16] Xing Yan, Chen Liang, Yize Jiang, Nana Luo, Zhou Zang, and Zhanqing Li. A deep learning approach to improve the retrieval of temperature and humidity profiles from a ground-based microwave radiometer. *IEEE Transactions on Geoscience and Remote Sensing*, 58(12):8427–8437, 2020.

[17] KR Knupp, T Coleman, D Phillips, R Ware, D Cimini, F Vandenberghe, J Vivekanandan, and E Westwater. Ground-based passive microwave profiling during dynamic weather conditions. *Journal of Atmospheric and Oceanic Technology*, 26(6):1057–1073, 2009.

[18] Kyle Hilburn. Inferring airmass properties from goes-r abi observations. http://dx.doi.org/10.1002/essoar.10504854.1, Nov 2020.

[19] Neerja Sharma and MM Ali. A neural network approach to improve the vertical resolution of atmospheric temperature profiles from geostationary satellites. *IEEE Geoscience and Remote Sensing Letters*, 10(1):34–37, 2012.

[20] Maohua Ding. A second generation of the neural network model for predicting weighted mean temperature. *GPS Solutions*, 24(2):1–6, 2020.

[21] Agie Wandala Putra and Chidchanok Lursinsap. Cumulonimbus prediction using artificial neural network back propagation with radiosonde indeces. In *Seminar Nasional Penginderaan Jauh*, page 153, 2014.

[22] Himadri Chakrabarty, CA Murthy, and Ashish Das Gupta. Application of pattern recognition techniques to predict severe thunderstorms. *International Journal of Computer Theory and Engineering*, 5(6):850, 2013.

[23] Francisco JL Lima, Fernando R Martins, Enio B Pereira, Elke Lorenz, and Detlev Heinemann. Forecast for surface solar irradiance at the brazilian northeastern region using nwp model and artificial neural networks. *Renewable Energy*, 87:807–818, 2016.

[24] Nina Håkansson, Claudia Adok, Anke Thoss, Ronald Scheirer, and Sara Hörnquist. Neural network cloud top pressure and height for modis. *Atmospheric Measurement Techniques*, 11(5):3177–3196, 2018.

[25] Mark S Veillette, Eric P Hassey, Christopher J Mattioli, Haig Iskenderian, and Patrick M Lamey. Creating synthetic radar imagery using convolutional neural networks. *Journal of Atmospheric and Oceanic Technology*, 35(12):2323–2338, 2018.

[26] Ryan Lagerquist, David Turner, Imme Ebert-Uphoff, Jebb Stewart, and Venita Hagerty. Using deep learning to emulate and accelerate a radiative-transfer model. *Journal of Atmospheric and Oceanic Technology*, conditionally accepted, 2021.

[27] Dan Li, Jianxin Zhang, Qiang Zhang, and Xiaopeng Wei. Classification of ecg signals based on 1d convolution neural network. In *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 1–6. IEEE, 2017.

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[29] Volodymyr Kuleshov, S Zayd Enam, and Stefano Ermon. Audio super-resolution using neural nets. In *ICLR (Workshop Track)*, 2017.

[30] B Schwartz and M Govett. A hydrostatically consistent north american radiosonde data base at the forecast system laboratory, 1946–present, noaa tech. *Memorandum, ERL, FSL-4, Boulder, Colorado*, 1992.

[31] Stanley G. Benjamin, Stephen S. Weygandt, John M. Brown, Ming Hu, Curtis R. Alexander, Tatiana G. Smirnova, Joseph B. Olson, Eric P. James, David C. Dowell, Georg A. Grell, Haidao Lin, Steven E. Peckham, Tracy Lorraine Smith, William R. Moninger, Jaymes S. Kenyon, and Geoffrey S. Manikin. A north american hourly assimilation and model forecast cycle: The rapid refresh. *Monthly Weather Review*, 144(4):1669 – 1694, 01 Apr. 2016.

[32] William Skamarock, Joseph Klemp, Jimy Dudhia, David Gill, Dale Barker, Wei Wang, Xiang-Yu Huang, and Michael Duda. A description of the advanced research wrf version 3. Technical report, National Center for Atmospheric Research, Mesoscale and Microscale Meteorology Division, 2008.

[33] Wan-Shu Wu, R. James Purser, and David F. Parrish. Three-dimensional variational analysis with spatially inhomogeneous covariances. *Monthly Weather Review*, 130(12):2905–2916, December 2002.

[34] Jeffrey S. Whitaker, Thomas M. Hamill, Xue Wei, Yucheng Song, and Zoltan Toth. Ensemble data assimilation with the NCEP global forecast system. *Monthly Weather Review*, 136(2):463–482, February 2008.

[35] Daryl T. Kleist, David F. Parrish, John C. Derber, Russ Treadon, Wan-Shu Wu, and Stephen Lord. Introduction of the GSI into the NCEP global data assimilation system. *Weather and Forecasting*, 24(6):1691–1705, December 2009.

[36] Oleg A. Alduchov and Robert E. Eskridge. Improved magnus form approximation of saturation vapor pressure. *Journal of Applied Meteorology (1988-2005)*, 35(4):601–609, 1996.

[37] Manuel SFV De Pondeca, Geoffrey S Manikin, Geoff DiMego, Stanley G Benjamin, David F Parrish, R James Purser, Wan-Shu Wu, John D Horel, David T Myrick, Ying Lin, et al. The real-time mesoscale analysis at noaa's national centers for environmental prediction: current status and development. *Weather and Forecasting*, 26(5):593–612, 2011.

[38] Wan-Shu Wu, R James Purser, and David F Parrish. Three-dimensional variational analysis with spatially inhomogeneous covariances. *Monthly Weather Review*, 130(12):2905–2916, 2002.

[39] Matthew T Morris, Jacob R Carley, Edward Colón, Annette Gibbs, Manuel SFV De Pondeca, and Steven Levine. A quality assessment of the real-time mesoscale analysis (rtma) for aviation. *Weather and Forecasting*, 35(3):977–996, 2020.

[40] Joseph J Charney, Shiyuan Zhong, Michael T Kiefer, Xiaoqing Zhu, Greg Soter, and Adam Cinderich. An investigation of the differences between real time mesoscale analysis and observed meteorological conditions at raws stations in the northeast united states. *JFSP Research Project Reports*, 25, 2013.

[41] Tim Schmit, Mat Gunshor, Gang Fu, Tom Rink, Kaba Bah, Wendy Zhang, and Walter Wolf. Noaa goes-r advanced baseline imager (abi) algorithm theoretical basis document for cloud and moisture imagery product (cmip). https://www.star.nesdis.noaa.gov/goesr/documents/ATBDs/Baseline/ATBD_GOES-R_ABI_CMI_KPP_v3.0_July2012.pdf, July 2012.

[42] Augustin Cauchy. Méthode générale pour la résolution des systemes d'équations simultanées. *Comptes Rendus de l'Academie des Science*, 25:536–538, 1847.

[43] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[44] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[45] Andrew Y Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78, 2004.

[46] NOAA National Centers for Environmental Information. Goes-r algorithm working group and goes-r series program: Noaa goes-r series advanced baseline imager (abi) level 2 clear sky mask. 10.7289/V5SF2TGP, 2018.

[47] Murry L. Salby. Fundamentals of atmospheric physics. *Academic Press*, page 627, 1996.

[48] David Bolton. The computation of equivalent potential temperature. *Monthly weather review*, 108(7):1046–1053, 1980.

[49] JM Wallace and PV Hobbs. Atmospheric science: An introductory survey. *Academic Press*, 467:350, 1977.

[50] Timothy J Schmit, Paul Griffith, Mathew M Gunshor, Jaime M Daniels, Steven J Goodman, and William J Lebair. A closer look at the abi on the goes-r series. *Bulletin of the American Meteorological Society*, 98(4):681–698, 2017.

[51] Andrew Heidinger and William Straka III. Noaa goes-r advanced baseline imager (abi) algorithm theoretical basis document for level 2 clear sky mask. https://www.star.nesdis.noaa.gov/goesr/docs/ATBD/Cloud_Mask.pdf, June 2013.