UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

# GRIDDED HAIL NOWCASTING USING UNETS, LIGHTNING OBSERVATIONS, AND THE WARN-ON-FORECAST SYSTEM

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

TOBIAS GARRICK SCHMIDT
Norman, Oklahoma
2023

Downloaded from SHAREOK. Persistent Handle URL: https://hdl.handle.net/11244/338834

GRIDDED HAIL NOWCASTING USING UNETS, LIGHTNING OBSERVATIONS, AND THE WARN-ON-FORECAST SYSTEM

A THESIS APPROVED FOR THE
SCHOOL OF METEOROLOGY

BY THE COMMITTEE CONSISTING OF

Dr. Amy McGovern, Chair

Dr. Corey Potvin, Co-Chair

Dr. John Allen

Dr. Cameron Homeyer

©Copyright by TOBIAS GARRICK SCHMIDT 2023
All Rights Reserved.

# Dedication

To my dearest parents and brother, who cared for and loved me throughout my life. Their love helped nurture a young scientist's desire to find some understanding of God's creation...

# Acknowledgements

Firstly, I would like to thank my advisors Dr. Amy McGovern and Dr. Corey Potvin for taking me on as their student and for supporting me throughout my time at OU. The expertise and knowledge I have gained from them will prove invaluable in future academic or industry endeavours. The many lighthearted moments (and stormshelters) we shared together will always be a source of fond memories for me. I also thank Dr. John Allen of Central Michigan University for generously donating his time each week to discuss thesis topics with a student he wasn't formally advising. I would also like to thank my mother Susan, my father Garry, and my brother Benjamin for all the support they have given me throughout my academic career. I must also thank my friends Joshua, Conner, Jarod, Alexander, and Nolan back in Canada who kept me encouraged through my degrees. Our many nights spent gaming while I was in the US and our travels when we could unite kept me pushing through. I specifically thank Nolan for the many machine learning discussions we had, which offered unique perspectives on my work. I of course must extend my thanks to Chad for his friendship and expertise on American culture during my time at OU. Thank you for joining me on numerous hair raising adventures, I look forward to the next ones. Studying severe weather at OU was a childhood dream and I must thank you all for making such a dream possible.

This material is based upon work supported by the National Science Foundation under Grant No. ICER-2019758. Funding was provided by NOAA/Office of Oceanic and Atmospheric Research under NOAA–University of Oklahoma Cooperative Agreement NA21OAR4320204, U.S. Department of Commerce. The computing for this project was performed at the OU Supercomputing Center for Education and Research (OSCER) at the University of Oklahoma (OU). The statements, findings, conclusions, and recommendations are those of the author(s) and do not necessarily reflect the views of NOAA or the U.S. Department of Commerce.

# Table of Contents

**Dedication** iv

**Acknowledgements** v

**List Of Tables** viii

**List Of Figures** ix

**Glossary** xiii

**Abstract** xv

**1 Introduction** 1

**2 Background** 5
- 2.1 Hail Forecasting . . . 5
- 2.2 Machine Learning, U-Nets, and Segmentation . . . 9
- 2.3 The Warn-on-Forecast System . . . 14

**3 Methods** 17
- 3.1 Datasets . . . 17
  - 3.1.1 Predictors . . . 18
    - 3.1.1.1 Warn-on-Forecast System . . . 19
    - 3.1.1.2 Vaisala Lightning Detection . . . 22
  - 3.1.2 Labels (GridRad MESH) . . . 24
  - 3.1.3 Other Data (GridRad Reflectivity) . . . 27
- 3.2 Data Mining and Pre-processing . . . 27
  - 3.2.1 Data Mining (AutoPatcher) . . . 28
  - 3.2.2 Preprocessing . . . 29
    - 3.2.2.1 Test Set Partition and Storm Day Clustering . . . 30
    - 3.2.2.2 Minima/Maxima and Architecture Data Slicing . . . 31
    - 3.2.2.3 Cross-Validation . . . 33
    - 3.2.2.4 Normalization and Remaining Preprocessing . . . 34
- 3.3 Machine Learning Architecture (U-Nets) . . . 35
- 3.4 Hyperparameter Searches and Model Training . . . 38
- 3.5 Max Critical Success Index Metric . . . 38

**4 Results** **41**
4.1 Critical Importance of Data Distribution . . . . . 41
4.2 Predictor and Label Experiments . . . . . 44
4.2.1 Gaussian Expansion Experiment . . . . . 46
4.2.2 Predicted Hail Size Experiment . . . . . 48
4.2.3 Lightning Observations Ablation Experiment . . . . . 49
4.3 Architecture Experiments . . . . . 53
4.4 Overall Performance . . . . . 56
4.5 Case Studies . . . . . 61
4.5.1 Case Study 1 (May 18, 2017) . . . . . 62
4.5.2 Case Studies 2 and 3 (May 19, 2018 and May 28, 2019) . . . . . 67

**5 Discussion and Conclusions** **76**

**Reference List** **80**

**Appendices** **89**
A.1 Further Data Distribution Discussion . . . . . 90
A.2 Early Experiments . . . . . 94
A.3 AutoPatcher Details . . . . . 99
A.4 Used Hyperparameters . . . . . 103

# List Of Tables

3.1 Machine learning data sources summarized. Details are described in subsections 3.1.1 and 3.1.2. . . . . 19

A.1 A table of the dates used for all dataset partitions. Cells highlighted in green indicate data drawn from the indicated dates either randomly or from using cross-validation, but always with the storm clustering system. Cells highlighted in blue indicate partitions sliced by the given dates with no additional cross-date shuffling. . . . . 91

A.2 A table of all hyperparameters used for the optimized U-Nets in all 3 architecture experiments. All architecture experiments are delineated by the white spacing. Architecture experiment 3 is the top, architecture experiment 2 is in the middle, and architecture experiment 1 is at the bottom. . . . . 104

# List Of Figures

2.1 A diagram outlining the general structure of a 2D U-Net. The data patches shown are for appearances only. For a diagram of the 3D U-Net most commonly used in this project, see Figure 3.3. Adapted with permission from Chase et al. 2023. . . . . 12

3.1 A diagram of the data slicing used in the first two architecture experiments. A legend is given at the bottom of the diagram. Each shape indicates a single timestep of the predictors. The colors indicate the time since the start of the machine learning forecast with the integers within each shape. Each dataset is indicated by a different type of shape. 20

3.2 A diagram of the data slicing used in the third architecture experiment. A legend is given at the right of the diagram. Each shape indicates a single timestep of the predictors. The colors indicate the time since the start of the machine learning forecast with the integers within each shape. Each dataset is indicated by a different type of shape. The primary difference in this diagram when compared to figure 3.1, is that a third spatial dimension has been placed on the y-axis indicating that time is resolved in an extra dimension as opposed to feature space. . . . 21

3.3 An overview diagram of the architecture for our best model. This overview is also a glimpse into what U-Nets look like in general. This model was a single 3D U-Net used in architecture experiment 3. All its convolutional layers are 3D and are displayed as such. This particular model had 2 convolutional layers per network level and skip-connections at each level because it is of the 3-plus variant. Each convolutional layer had 8 5x5 kernels, batch normalization, and a ReLU activation function. All hyperparameters for this model and all others produced are available in Table A.2 . . . . . 36

3.4 A flowchart of the entire data mining and preprocessing pipeline. Each dark blue square is a software step. Each light blue rectangle or square is data. Data shapes are not to scale. The green shades help delineate the 3 architecture experiments which are each labeled. All 3 architecture experiments have similar high-level structure, however architecture experiment 2 has to repeat its steps 9 times (as indicated by the red boxed section) to resolve all timesteps in our forecast period. Most steps in this diagram are discussed throughout chapter 3. The left most Auto-Patcher code box is additionally described in Figure A.1, while each of the slicing code blocks are additionally described in Figures 3.1 and 3.2. 40

4.1 Results of an experiment indicating the relationship between non-observation-containing machine learning models and NWP performance. All points are pixelwise max CSI. The CSIs are a function of time since WoFS initialization. The red lines indicate test datasets that are unshuffled across time while the blue lines indicate test datasets that are shuffled completely with the storm clustering system. The lines with triangle markers indicate a surrogate for WoFS convection performance (CSIs from comparison of WoFS composite reflectivity >40 dBZ against GridRad composite reflectivity >40 dBZ) while the lines with circle markers represent the corresponding machine learning performance. . . . . 42
4.2 The results of our first predictor/label experiment. This experiment removed/added the Gaussian-based label expansion used in our train and validation sets. The left plot displays results using a pixelwise max CSI while the right plot uses max CSI with a 6 km radius neighborhood. Both plots examine the CSIs as a function of time since the ML forecast start. The black lines are the baseline cases, the red lines are the non-Gaussian models, and the blue lines are the original Gaussian models. . 46
4.3 The results of our second experiment. This experiment examined the effects of changing our predicted hail size from 1 inch to 0.3937 inches (10 mm). The left plot displays results using a pixelwise max CSI while the right plot uses max CSI with a 6 km radius neighborhood. Both plots examine the CSIs as a function of time since the ML forecast start. The black lines are the baseline cases, the red lines are the any-hail models, and the blue lines are the original severe hail models. . . . . 48
4.4 The results of our final ablation experiment. This experiment removed the lightning observations from the model's predictors. The left plot displays results using a pixelwise max CSI while the right plot uses max CSI with a 6 km radius neighborhood. Both plots examine the CSIs as a function of time since ML forecast start. The black lines are the baseline cases, the red lines are the ablation models, and the blue lines are the original models. . . . . 50
4.5 The results of comparing our three architecture experiments. The left plot displays results using a pixelwise max CSI while the right plot uses max CSI with a 6 km radius neighborhood. Both plots examine the CSIs as a function of time since the ML forecast start. The black lines are the usual baseline cases, the yellow lines are the 2D U-Net baseline models (architecture experiment 1), the red lines are from the cluster of time-resolving 2D U-Nets (architecture experiment 2), and the blue lines are the best performing models produced (the 3D U-Nets of architecture experiment 3). . . . . 53
4.6 A diagram indicating pixelwise reliability for our best model at 15 minute intervals of the forecast period. The center dashed line indicates ideal performance. . . . . 57

4.7 A diagram indicating model performance using area under the curve for our best model at 15 minute intervals of the forecast period. The closer a line is to the top left corner of the plot, the better its model performance. 58
4.8 A map of western Oklahoma and northwestern Texas on May 18, 2017 starting at 19:15 UTC. A 45 minute swath of the forecasted probability of severe hail produced by our best model (ML output) is overlaid in black contours. HAILCAST >1 inch from a single WoFS ensemble member is displayed in blue. GridRad MESH >1 inch is displayed in red. Any significant or non-significant severe hail report that occurs during this 45 minute period is displayed with a black star or black circle respectively. 63
4.9 3 maps of western Oklahoma and northwestern Texas on May 18, 2017 at 19:15 UTC, 19:30 UTC, and 19:55 UTC. This figure is made up of 3 distinct types of maps (the rows) with 3 shown timesteps (the columns) in the forecast period, all past the first valid timestep. The first map (top row) displays WoFS composite reflectivity from 1 ensemble member with our best model's predicted probability of severe hail overlaid as black contours ranging from 0 to 1.0. The second map (middle row) displays GridRad composite reflectivity with our best model's predicted probability of severe hail overlaid as black contours ranging from 0 to 1.0. The final map (bottom row) displays our best model's predicted probability of severe hail against the other two hail surrogates. All storm reports in each map are from the slice of time between 10 minutes before the timestep to 10 minutes after. . . . . 64
4.10 A map of northern Oklahoma on May 19, 2018 starting at 19:15 UTC. A 45 minute swath of the forecasted probability of severe hail produced by our best model (ML output) is overlaid in black contours. HAILCAST >1 inch from a single WoFS ensemble member is displayed in blue. GridRad MESH >1 inch is displayed in red. Any significant or non-significant severe hail report that occurs during this 45 minute period is displayed with a black star or black circle respectively. . . . . 67
4.11 3 maps of northern Oklahoma on May 19, 2018 at 19:15 UTC, 19:30 UTC, and 19:55 UTC. This figure is made up of 3 distinct types of maps (the rows) with 3 shown timesteps (the columns) in the forecast period, all past the first valid timestep. The first map (top row) displays WoFS composite reflectivity from 1 ensemble member with our best model's predicted probability of severe hail overlaid as black contours ranging from 0 to 1.0. The second map (middle row) displays GridRad composite reflectivity with our best model's predicted probability of severe hail overlaid as black contours ranging from 0 to 1.0. The final map (bottom row) displays our best model's predicted probability of severe hail against the other two hail surrogates. All storm reports in each map are from the slice of time between 10 minutes before the timestep to 10 minutes after. . . . . 68

4.12 A map of central Oklahoma on May 28, 2019 starting at 22:45 UTC. A 45 minute swath of the forecasted probability of severe hail produced by our best model (ML output) is overlaid in black contours. HAILCAST >1 inch from a single WoFS ensemble member is displayed in blue. GridRad MESH >1 inch is displayed in red. Any significant or non-significant severe hail report that occurs during this 45 minute period is displayed with a black star or black circle respectively. . . . . 69
4.13 3 maps of central Oklahoma on May 28, 2019 at 22:45 UTC, 23:00 UTC, and 23:15 UTC. This figure is made up in the same format as figure 4.9. 70
4.14 A map of northern Kansas on May 28, 2019 starting at 22:45 UTC. A 45 minute swath of the forecasted probability of severe hail produced by our best model (ML output) is overlaid in black contours. HAILCAST >1 inch from a single WoFS ensemble member is displayed in blue. GridRad MESH >1 inch is displayed in red. Any significant or non-significant severe hail report that occurs during this 45 minute period is displayed with a black star or black circle respectively. . . . . 71
4.15 3 maps of northern Kansas on May 28, 2019 at 22:45 UTC, 23:00 UTC, and 23:15 UTC. This figure is made up in the same format as figure 4.9. 72

A.1 A flowchart of the autopatcher process. Each square is a primary software step. The green colors in each of the expanded sections indicate timesteps that are NWP initialization times. Number in that section indicates how many minutes the datasets are offset by to make the matches. The red boxes indicate successful matches. Each star is a conditional logic step. All steps are numbered and are accordingly explained throughout 3.2.1 . . . . . 100

# Glossary

**AI2ES** NSF AI Institute for Research on Trustworthy AI in Weather, Climate, and Coastal Oceanography

**AUC** Area Under the ROC Curve

**CAPE** Convective Available Potential Energy

**CIN** Convective Inhibition

**CNN** Convolutional Neural Network

**CSI** Critical Success Index

**DBSCAN** Density-Based Spatial Clustering of Applications with Noise

**ESMF** Earth System Modelling Framework

**FSS** Fractions Skill Score

**GRAF** Global High-Resolution Atmospheric Forecasting System

**IID** Independent and Identically Distributed

**ISO** International Organization for Standardization

**LFC** Level of Free Convection

**LCL** Lifting Condensation Level

**LSTM** Long Short-Term Memory

**MESH** Maximum Expected Size of Hail

**ML** Machine Learning

**NLDN** National Lightning Detection Network

**NSF** National Science Foundation

**NWP** Numerical Weather Prediction

**ROC** Receiver Operating Characteristic

**SHI** Severe Hail Index

**WoFS** Warn-on-Forecast System

**WRF** Weather Research and Forecasting Model

# Abstract

Hailstorms cause around 1 billion dollars in damage across the United States each year. At least a portion of this cost is associated with the inability to protect personal assets from damage in the short window of time offered by a severe weather warning. To address this problem, we developed a nowcasting model that uses UNet style convolutional neural networks (CNNs) to produce gridded severe hail forecasts for the next hour. One of the advantages of machine learning models is their ability to fuse large quantities of data from traditionally disparate sources such as ground observations and model output to produce a forecast. To exploit this hybrid predictor potential, these models are trained on the high-resolution (3 km spatial, 5 min temporal) output from the Warn-on-Forecast System (WoFS) numerical weather prediction (NWP) ensemble and remote sensing observations from Vaisala's NLDN lightning detection system. Maximum expected size of hail (MESH) from the gridded NEXRAD WSR-88D radar (GridRad) dataset is used as the model's truth labels. In addition to traditional machine learning optimization techniques such as hyperparameter searches and predictive feature selection, several different UNet architectures are compared to obtain a better machine learning model. The high-resolution nature of this data enables strategies such as using time as an additional dimension in a 3D UNet. This 3D model is compared against the effectiveness of a traditional 2D UNet. Finally, both models are compared against HAILCAST and simple logistic regression trained on 2 to 5 km updraft helicity to investigate their effectiveness.

# Chapter 1

# Introduction

The accurate forecasting of hail has significant implications for the protection of life and property. Hailstorms often cause billions of dollars in damage each year in the US (Gunturi and Tippett 2017). On April 28, 2021 a single hailstorm passing over Norman, Oklahoma caused one billion US dollars in damage (NSSL 2021).

Hail forecasting is of particular value to the insurance industry as hail can be forecasted to reduce possible damage (and thus reduce claims) and hail forecasting can help improve insurance risk calculations (Changnon et al. 1997). For example, Tomorrow.io reported that weather forecasts can save an average of $3,000 per hail claim when their clients' users had access to their data (Beauchemin 2023). This adds considerable value to the prediction of this particular severe weather hazard, as insurance companies have a motive to encourage damage mitigation whenever possible. Additionally, even if a consumer has insurance, hail damage is disruptive to their personal lives as repairs can take months. Even lead time within an hour is beneficial to the general public, so it was decided to focus on this time frame for this thesis. This first hour of forecasting is called "nowcasting".

The pursuit of accurate and precise hail nowcasting has been ongoing for many years e.g. (Foster and Bates 1956, Brimelow et al. 2002, Adams-Selin and Ziegler 2016, Gagne et al. 2017, Adams-Selin et al. 2019). Throughout the history of this endeavor many of the core obstacles to this objective have remained the same. One such obstacle is the spatial and temporal scale at which hail forms. The formation of hail (both severe and

non-severe) within an updraft is a small-scale process that cannot be directly resolved at the coarser resolutions of current numerical weather prediction models (Hong and Dudhia 2012, Yano et al. 2018). Due to the computational limitations of computers, it will be impossible to resolve these scales for the foreseeable future. Thus, a surrogate for these processes is required.

Physics-based statistical models such as HAILCAST (Jewell and Brimelow 2009, Adams-Selin and Ziegler 2016) are a popular choice for use as a surrogate. However, in more recent years, machine learning has been used to accomplish this goal e.g. (Gagne et al. 2017, McGovern et al. 2017, Gagne et al. 2019, Czernecki et al. 2019, Lagerquist et al. 2020, McGovern et al. 2023). In general, machine learning has been shown to perform well for this task (Gagne et al. 2017, Lagerquist et al. 2020, Sha et al. 2020, Kochkov et al. 2021). Machine learning is known for determining statistical relationships within large data problems such as weather forecasting better than humans e.g. (Gagne et al. 2017, Lagerquist et al. 2020, Sha et al. 2020, Kochkov et al. 2021). For this reason, nowcasting systems made from machine learning models were an ideal candidate to assist in the prediction of hail from coarse datasets. Specifically, this thesis focuses on their use for the probabilistic nowcasting of severe hail (hail with a diameter $\geq 1$ inch).

Our primary objective was to develop an effective severe hail nowcasting model using machine learning. Effectiveness was measured by comparing our model's performance against the performance of well-established hail prediction baselines and other machine learning methods. To create our hail model, we specifically selected a family of machine learning models known as U-Nets (Ronneberger et al. 2015, Özgün Çiçek et al. 2016, Huang et al. 2020). U-Nets were chosen as they have been shown to perform well in many recent severe nowcasting tasks (Lagerquist et al. 2020, Sha et al. 2020, Justin et al. 2023). This can be partially attributed to the skill of U-Nets when

training on spatial data, which is a type of data commonly used for meteorological problems. They often use images (or slices of data) as input and return output for each pixel. For our purposes, each of these pixels will be returned as the probability of severe hail. This is an ideal format for representing spatial geophysical data because it is most similar to what a forecaster would consume in a real operational environment.

Many severe weather nowcasting/forecasting studies have found success in exclusively using real-time observations as the input predictors for a machine learning model e.g. (Billet et al. 1997, Manzato 2013, Lagerquist et al. 2020, Gensini et al. 2021). Others have used numerical weather prediction (NWP) model output to create their input datasets (Gagne et al. 2017, McGovern et al. 2023). However, it is less common for researchers to combine both these methods together, despite examples of success when using a hybrid approach (Czernecki et al. 2019). This approach is attractive as it theoretically exploits the convective accuracy of real-time observations, while still using the accurately forecasted portions of NWP models, such as their environmental variables. As such, we elected to attempt the hybrid approach with a combination of high-resolution US lightning observations supplied by Vaisala (Pohjola and Mäkelä 2013) and the high-resolution Warn-on-Forecast System (WoFS) NWP model ensemble (Stensrud et al. 2009, Stensrud et al. 2013). One of the reasons both of these datasets were selected was because they are of similar scale to two global data sources: IBM's Global High-Resolution Atmospheric Forecasting System (GRAF) model and Vaisala's global lightning dataset. If a machine learning model was effective on these smaller-scale datasets, it may be effective on their global counterparts. A secondary objective of this thesis was to measure the benefits of switching from an exclusively NWP sourced set of inputs to one which also included the lightning observations. It was hypothesized that this would result in a significant increase in performance, and

if so could argue for the effectiveness of a similar AI-driven hail model trained on the associated global lightning observations.

Finally, to increase the likelihood of finding an effective machine learning model for use in hail nowcasting, we performed 3 distinct architecture experiments that all used various U-Nets. Our final objective was to measure the performance differences across these architecture experiments to determine which was the best. With experiments in both data selection and architecture selection, it was believed that any produced hail nowcasting system could be reliably vetted.

# Chapter 2

# Background

I begin with a background discussion of the broad concepts used and referenced throughout the thesis. This chapter is broken into 3 sections. The first discusses hail and related topics. The second discusses machine learning broadly along with details regarding the specific model we used. Finally, the third section discusses the numerical weather prediction modelling system that was used for some of our machine learning predictors, called the Warn-on-Forecast System.

## 2.1 Hail Forecasting

The exact physical processes behind hail formation and growth continue to be debated amongst leading hail experts. What is generally accepted is that hail formation begins with the development of a hail embryo that subsequently moves through a storm updraft. Eventual hail diameter is a function of the time this embryo spends inside the storm updraft. If an abundance of supercooled water is present (which is not always the case), the water will freeze to the edges of the hailstone during its time spent in the updraft. These concepts are well summarized in (Allen et al. 2020). Generally, the longer a hailstone remains in the updraft, the larger the hailstone becomes.

These processes are part of a set of complex interactions referred to as microphysics (Labriola et al. 2019, Allen et al. 2020, Morrison et al. 2020). These are physical processes that occur at scales far less than 1 meter (Morrison et al. 2020). Our computational limits keep operational NWP model grid cells at the scale of a few kilometers at their smallest (operational NWP examples: Stensrud et al. 2009, Stensrud et al. 2013). This greatly exceeds the scale necessary to resolve these microphysics. As such, most current methods for the forecasting/nowcasting of hail must use a statistics-based or physical surrogate of these processes that would broadly occur within a single NWP model grid cell (Milbrandt and Yau 2005, Stensrud et al. 2009, Stensrud et al. 2013, Labriola et al. 2019). As a result, our current hail forecasts based on this system cannot produce high quality predictions beyond a fixed limit (Morrison et al. 2020). Additionally, the microphysics that occur are numerous and contain many competing processes (Morrison et al. 2020). Each of these processes offer a certain percentage of the total physical explanation of the hail growth model and one can debate their individual degrees of relative importance owing to limitations of direct observations (Morrison et al. 2020).

Despite these difficulties, many different statistical and physics-based hail models have found success in forecasting hail for various lead times. Some methods worked by producing hail forecasts exclusively based upon environmental variables, while other methods worked by assuming ongoing convection and then estimating possible hail growth within said convection. Producing methods based exclusively around environmental variables can prove difficult as our understanding of these environmental variables is still partially incomplete, just as with our knowledge of hail microphysics (Allen et al. 2020). For example, certain environments support rapid hail growth where hail can spend less time in a storm updraft, while other environments support slower hail growth with greater time required in an updraft (Allen et al. 2020). This has

resulted in many different physics-based approaches to hail forecasting using environmental variables.

One set of examples of these environment-based methods are the methods produced by directly relating soundings from observations/models to hail reports. A study in Finland on near storm environments from observation-based soundings determined that storm mode forecasting could form the basis for stronger forecasting of hail in Finland (Tuovinen et al. 2015). Another study used model analysis-sourced soundings to evaluate the effectiveness of many common meteorological variables when forecasting hail of different sizes (Johnson and Sugden 2014). This study also emphasized the importance of better forecasting storm mode to assist in the use of environmental variables for forecasting hail, among other findings. In general, the broadest methods for forecasting/nowcasting hail from environmental sounding-based variables tended to be focused around the forecasting of severe weather conditions, as opposed to direct hail forecasting (Thompson et al. 2003, Allen et al. 2011). These solutions tended to offer mixed success (Allen et al. 2020), however microphysics still play an important role in the forecasting of storm mode itself, and thus the NWP scaling issue still hampers the effectiveness of these methods (Smith et al. 2012).

A popular subset of environment-based methods that has proven effective in hail forecasting are the methods focused on the prediction of updraft strength. Strong updrafts are believed to be necessary for hail growth, so forecasting their strength is logical (Allen et al. 2020). Some studies focused on the use of Convective Available Potential Energy (or CAPE) as a surrogate for updraft strength (Tuovinen et al. 2015, Púčik et al. 2015). Others use alternative surrogates such as the lifted index (Mohr and Kunz 2013). Regardless of the source of these surrogates, methods focused on updraft strength have shown skill in numerous case studies, and in particular they have proven popular in European applications (Allen et al. 2020). However, one possible limitation

to these methods is that more recent studies have suggested that updraft width may play a more significant role on hail growth than updraft strength (Nelson 1983, Dennis and Kumjian 2017). This is because it is possible updraft width is a greater driver for the time hail spends in an updraft, which is an established requirement for hail growth.

An alternative to these purely environment-based methods are methods that assume convection in some manner is occurring and then estimating possible hail growth through physics-based estimates of the time hail would spent in the updrafts of said convection. For many years, a popular example of one of these methods has been HAILCAST (Brimelow et al. 2002, Jewell and Brimelow 2009, Adams-Selin and Ziegler 2016). HAILCAST works by using environmental soundings as input for an ensemble of simulated updrafts formed from slight perturbations of its initial conditions (Jewell and Brimelow 2009, Adams-Selin and Ziegler 2016). If the parcels within an ensemble member are capable of passing their level of free convection, they are considered to be part of an active updraft and hail size estimates are produced by injecting a simulated hail embryo into the updraft. The final hail results are then made from the ensemble statistics of the expected growth size of each of these embryos. HAILCAST was shown to perform well during the 2014–2016 NOAA Hazardous Weather Testbeds and has since become commonly used as a hail prediction tool (Adams-Selin et al. 2019). In particular, the latest version of this method (Adams-Selin and Ziegler 2016) is used within some convection-allowing models such as the Warn-on-Forecast System, which is the modeling system used for our NWP predictors as discussed below. HAILCAST serves as an excellent product for comparison against any produced hail forecasting/nowcasting models.

## 2.2 Machine Learning, U-Nets, and Segmentation

Despite the successes of physics-based methods for hail forecasting/nowcasting, in more recent years a popular method for creating hail models has been machine learning. Machine learning has seen an explosion of uses throughout the broader meteorological community in this same time period (Chase et al. 2022). Machine learning has the potential to be a valuable alternative to physics-based methods because machine learning is excellent at finding complex relationships necessary to solving large-data problems. The nowcasting of hail is one such complex large-data problem. Examples of successful machine learning uses in hail forecasting and nowcasting include: the use of random forests, logistic regression, and k-means clustering for all hazards severe weather prediction explored by McGovern et al. 2017, next-day hail forecasting using random forests trained on NWP models by Gagne et al. 2017, and Gagne et al. 2019 used convolutional neural networks to help identify hail inside radar reflectivity observations. Further examples cited in this thesis include Czernecki et al. 2019 and Lagerquist et al. 2020, but a more complete overview can be seen from (McGovern et al. 2023) as there are numerous examples.

In general, machine learning is capable of determining its own connections between complex inputs (often referred to as predictors) and outputs (often referred to as labels or truth). For the models used in this project, the machine learning finds these connections through a process known as "training" where the model optimizes a series of internal numerical weights using a predefined training dataset. The resulting weights make up the model's "solution" to the problem, and the model can then be used on data it has not seen before to create useful output.

Machine learning tasks tend to fall into one of two categories. The first category is referred to as supervised learning. Supervised learning is when the machine learning

model possesses both the predictors and labels necessary to determine any connecting relationships. The second category is called unsupervised learning. In this method, the machine learning model must determine any details regarding its output during training without the assistance of labels. For an excellent overview of this process, see (Chase et al. 2022). For our application, we elected to use supervised learning to accomplish our goals. As such, we required both predictor and label datasets. These predictors were NWP environmental data and real-time observations, while our labels were a hail surrogate derived from radar observations.

The type of machine learning model we selected for this project is from a popular family of models known as neural networks (Wang and Raj 2017). Neural networks consist of large collections of "neurons" which themselves consist of mathematical activation functions and connections that facilitate the passing of data (Picton 1994, Wang and Raj 2017). Activation functions work by returning a binary signal when a series of inputs cross a defined threshold. This return signal can then be used by any following functions. These "neurons" made up of activation functions and their various connections are then merged in a network with other "neurons" in a dense fashion similar to a tree-like data structure (Picton 1994, Wang and Raj 2017). The resulting network becomes a non-linear problem capable of describing a large set of applications (Picton 1994, Wang and Raj 2017). These activation functions and connections are akin to activation signals and axons of a human neuron, hence these networks are referred to as neural networks (Picton 1994).

A particularly popular form of neural network for problems that contain spatial data is the convolutional neural network or CNN (Albawi et al. 2017). This form of neural network uses small filters known as kernels to translate spatial features (for example, a spatial wind field gradient) into real numbers that can be fed through the neurons of a neural network (Albawi et al. 2017). The output of these filters and their neurons are

then convolved into a smaller set of spatial floats that can be passed through another set of filters (Albawi et al. 2017). These steps together can be described as a convolutional layer. This process is repeated many times until the final output of a single scalar real number is produced at the end of the network (Albawi et al. 2017).

The particular machine learning problem that had to be solved for this project is referred to as segmentation (Ronneberger et al. 2015). Segmentation is when a machine learning model produces output for multiple pixels in a higher dimensional image as opposed to a single dimensional output (Ronneberger et al. 2015). As such, segmentation is an excellent fit for a meteorology problem where output is required to represent many pixels in a coarse grid domain. Traditional CNNs are not designed to solve problems of this nature. As discussed, what is returned at the bottom of a traditional neural network (or what could be described as the network output) is usually a single scaler or a one-dimensional array of numbers (Albawi et al. 2017). In a segmentation problem, every pixel (or data point) within an image (or other two-dimensional data structure) must receive a float output (that then can be used in pixelwise classification if desired) as opposed to a single float (Ronneberger et al. 2015).

For this project, we elected to use a type of neural network referred to as a U-Net (Ronneberger et al. 2015, Özgün Çiçek et al. 2016, Huang et al. 2020). For an example of a U-Net, see Figure 2.1. U-Nets were specifically created to solve this segmentation task and were thus ideal for this project (Ronneberger et al. 2015, Özgün Çiçek et al. 2016, Huang et al. 2020). A traditional two-dimensional U-Net consists of multiple downsampling layers of convolutions and filters followed by multiple upsampling layers (Ronneberger et al. 2015, Huang et al. 2020). This downsampling direction followed by upsampling direction when displayed in a diagram is visually indicative of the

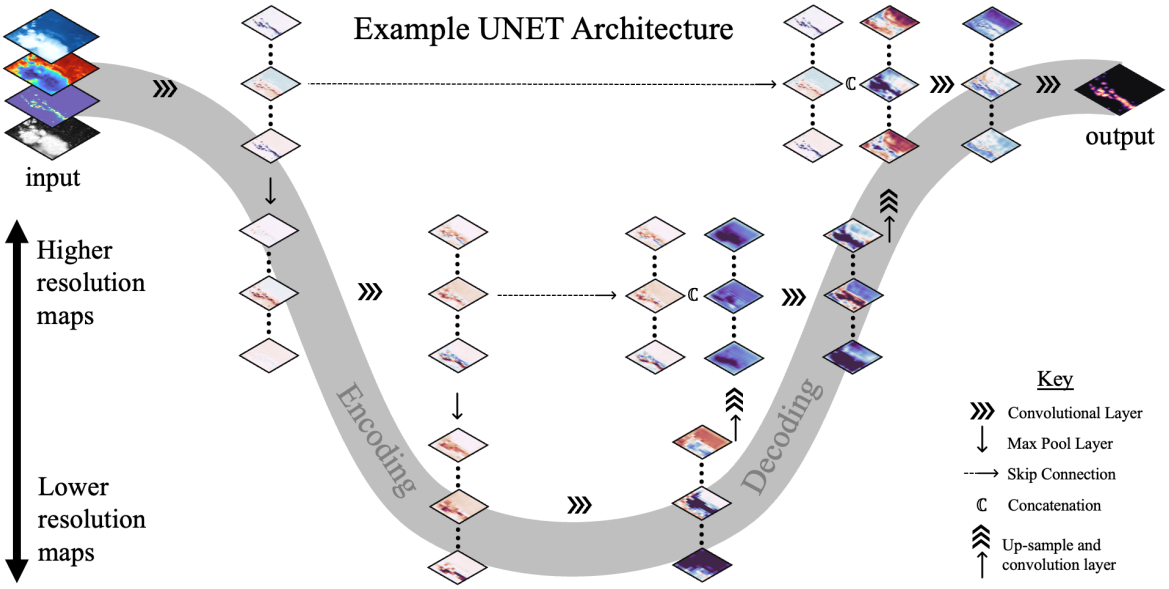



Figure 2.1: A diagram outlining the general structure of a 2D U-Net. The data patches shown are for appearances only. For a diagram of the 3D U-Net most commonly used in this project, see Figure 3.3. Adapted with permission from Chase et al. 2023.

letter "U", hence, the name U-Net (Ronneberger et al. 2015, Huang et al. 2020). Additionally, some more recent versions of U-Nets contain skip connections which are network connections between downsampling and upsampling layers of the same depth (Huang et al. 2020). As discussed, the convolutional layers used during downsampling reduce the dataset's size with each layer. This process is repeated some number of times defined by the depth of the U-Net given before the start of the experiment. This shrinking of the data's shape is the basis of the term downsampling and when the bottom of the "U" is reached the data's shape is much smaller relative to its starting shape.

Throughout this whole process, weights are applied to each filter at each level (Ronneberger et al. 2015, Albawi et al. 2017). These weights make up the parameters the machine learning optimization algorithm will attempt to optimize based upon a metric

of modeling success called a loss function. In the case of neural networks, the optimization of the loss function is a non-linear problem because, as discussed previously, the entire network itself is a non-linear collection of functions (Albawi et al. 2017). Therefore, a more complex optimization algorithm that exploits backpropagation such as Adam (Kingma and Ba 2017) must be used in place of straight stochastic gradient descent.

Once the data has passed down to the lowest level of the network and has completed downsampling, the weights become one-dimensional and thus require several increases in dimension such that they can represent the final output shape needed for segmentation. Upsampling accomplishes this task throughout the second half of the network. In order for the output shape to be the same as the input shape (and therefore correctly perform segmentation), the same number of layers must be used in the upsampling phase as what was used in the downsampling phase (Ronneberger et al. 2015). Each upsampling layer has a similar structure to each downsampling layer, however, as opposed to containing convolutions that reduce the size of the data's shape, each upsampling layer contains an upsampling function that increases the size of the data's shape (Ronneberger et al. 2015). These functions are often as simple as a function that copies each passed data point n number of times, where n is the factor of shape size reduction that occurs in each upsampling layer's corresponding downsampling layer (Ronneberger et al. 2015). Additionally, each upsampling layer also contains their own weights used in the optimization problem such that further relationships can be determined towards the end of the network. These weights are functionally equivalent to a convolutional layer made up of filters with a 1 x 1 shape. If the U-Net contains skip connections, these will be inputs for each of the upsampling layers. These connections would originate from the output of each downsampling layer.

They allow for the model to learn relationships that do not require the full depth of the network to be sufficiently resolved.

The end result of any U-Net as described above is a model that translates an image-like data structure with any number of features (or channels) into a similarly shaped image-like data structure. For our application, the final output is pixelwise probabilities of severe hail.

## 2.3 The Warn-on-Forecast System

As was briefly summarized, our machine learning-based nowcasting solution exploits a hybrid NWP/observation approach to predictors. This was to allow our machine learning models to exploit the advantages of an already established forecast while having observation ground truths to help control any NWP error that would be expected when trying to resolve convection. Furthermore, at the timescale of nowcasting, we expected NWP to not lose too much of its accuracy in contrast to longer-ranged forecasting, especially with regards to its environmental variables.

This successful machine learning model was produced by selecting a NWP model well-suited for the nowcasting timescale. This model would also require the highest spatial resolution possible to optimally resolve at least some mesoscale storm features while still being capable of running quick enough for real-time operations. High resolution convection-allowing-models have been shown to resolve some mesoscale features with grid spacing less than or equal to 3 km (Potvin and Flora 2015). An well-suited NWP model for these constraints is the experimental Warn-on-Forecast System (WoFS) model ensemble (Stensrud et al. 2009, Stensrud et al. 2013, Gallo 2017). The WoFS model ensemble was originally envisioned in 2009 as high-resolution operational severe weather modeling system that could assist in the National Weather Service warning

process (Stensrud et al. 2009) The primary goal of WoFS is to integrate its forecast directly into the warning process (Stensrud et al. 2013). In addition to a NWP model ensemble, WoFS is made up of a rapidly-refreshed data assimilation system which is used to greatly increase model performance and reliability at the shorter lead times (Hu and Xue 2007, Stensrud et al. 2013). It was believed that this NWP modelling system could predict its own storms due to its rapid data assimilation, as opposed to just offering future environmental considerations to a forecaster's analysis of an already observed mesoscale storm (Stensrud et al. 2009).

In contrast to most NWP models which have a fixed forecast domain (such as CONUS or regional models), WoFS has a smaller high-resolution domain that is positioned over areas in the United States where severe weather is expected to occur. During operations, WoFS is initialized every 30 minutes. It rapidly outputs data every 5 minutes and performs rapid data assimilation at a rate of every 15 minutes. Its data assimilation system is built on the 3DVAR system which effectively translates 3D radar data into storm information for model ingestion (Gao 2013, Calhoun et al. 2014, Smith 2014). In particular, this data assimilation system has a high cycling rate (5 minutes) which allows detailed convective information to enter WoFS in a timely manner, which is essential in convection timescales. Its most recent specifications include a spatial grid resolution of 3 km by 3 km made of of 300x300x50 pixel domains. In previous version of WoFS the x-y dimensions were only 250x250 pixels, including during some of the years used for this thesis. As such, for our purposes, efforts had to be made in order to ensure sampling could adapt to different domain sizes. Generally, these specifications made WoFS output a clear choice for use as predictors in a machine learning model. In particular, the resolution of the Warn-on-Forecast System and its rapid data assimilation have shown success in the nowcasting of severe weather in the past, including in machine learning applications (Flora et al. 2021). Additionally, the 3x3 km grid

spacing of WoFS is the same spatial scale of IBM's GRAF model over land. Ideally, this would imply that the performance of a model trained on WoFS data would be similar to the performance of what could be trained on a global high-resolution model like GRAF.

Several fields from WoFS will be used as predictors for our machine learning models. These are outlined later in the Methods chapter. One field of particular importance is the HAILCAST field (Adams-Selin and Ziegler 2016), as introduced above. This field is both used as a predictor for our models and is used as a baseline for basic hail forecasting performance. HAILCAST is a commonly used surrogate for hail and is thus an excellent choice for a baseline. It performs well on its own, but is limited by the constraints of physics-based interpretations of microphysics, which cannot be resolved at current NWP scales. As such, HAILCAST is useful to act primarily as a physics-based alternative to our machine learning results when comparing model performances in the Results chapter (chapter 4).

# Chapter 3

# Methods

This chapter will discuss our particular uses of the concepts introduced in the background chapter. It is divided into 5 sections. The chapter begins with a section detailing all the datasets used throughout the thesis. This is broken into subsections depending on whether the data is used as machine learning predictors or truth labels. The second section discusses the data mining process and the following preprocessing steps. Thirdly, the machine learning we used in the thesis is discussed in more detail, including how it relates to different machine learning architectures we experimented with. The fourth section discusses the training process itself along with how hyperparameter searching was performed. Finally, the fifth section outlines the primary metric used for performance evaluation in this thesis, the critical success index, along with its variations.

## 3.1 Datasets

The datasets used in this thesis are made up of remote sensing observations and numerical weather prediction model output. The observations are sourced from the NEXRAD US national doppler radar network and ground-based lightning sensors. The numerical weather prediction data is exclusively from the Warn-on-Forecast System, which was introduced earlier. All data sources are divided into sections depending on if they contributed to our predictors or the truth labels.

### 3.1.1 Predictors

The first datasets that will be discussed are the predictors. Machine learning projects throughout the field of meteorology use a wide range of predictors that can be categorized in different ways. For the purposes of this thesis we will categorize them by how they drive the forecasting work performed within the machine learning model. Using this framework, most predictors can be put into one of two categories. The first category is described as the predictors made up of meteorological observations or analyses. Predictors in this category have the advantage of being high-quality, high-resolution, and *relatively* (when compared to NWP) low-bias data that often can describe storms in intricate detail. The second category is data made up of output fields from NWP models. Predictors from this source do have their own forecasting information built-in and Machine learning models built on this data already have access to an estimate of future storm information before training even begins.

Predictors that originate from NWP allow the machine learning model to learn associative processes with NWP output without having to determine the entire predictive process on its own. Therefore, some advantages one would expect from this include a reduced amount of time required for training, and requiring less training data. It has been consistently observed throughout this thesis that the rarity of hail occurrences greatly increased the difficulty of all attempted machine learning tasks, thus any advantage that combats this is useful. Derived fields are primarily used for our predictors because the same amount of information contained within many raw fields is contained within few derived fields and less predictors allows the machine learning to find a solution easier. However, a downside to NWP based predictors is that the machine learning model can "over trust" in the accuracy of the NWP output. For example: it is possible that a machine learning model puts too much emphasis on the composite reflectivity field of a NWP model. This can occur because a partially

minimized saddle point found during optimization is likely present with a model that can at least perform reasonably well when assuming an effective linear relationship between NWP reflectivity and severe weather. Thus, as with many machine learning tasks, much of the work done when using these predictors is spent tuning the machine learning such that it avoids these simpler solutions.

#### 3.1.1.1 Warn-on-Forecast System

**Predictors**

| NWP (Warn-on-Forecast System) | | | | |
|---|---|---|---|---|
| Convection | Severe Weather Composites | Wind Shear | Humidity | Sounding Heights |
| • W-up<br>• MU CAPE<br>• MU CIN<br>• SFC CAPE<br>• SFC CIN | • Hail (WRF)<br>• Hailcast<br>• SCP | • UH 2 - 5 km<br>• SRH 0 – 1 km<br>• SRH 0 – 3 km<br>• V-Shear 0 – 6 km<br>• U-shear 0 – 6 km | • $T_d$ (2 m) | • Freezing Level<br>• MU LFC<br>• MU LCL |

| Observations (Vaisala Lightning Network) |
|---|
| • Lightning Event Count per 3 x 3 km Bin |

**Truth**

| GridRad (NEXRAD WSR-88D Radar Network Data) |
|---|
| • Maximum Expected Size of Hail (MESH) |

Table 3.1: Machine learning data sources summarized. Details are described in subsections 3.1.1 and 3.1.2.

As mentioned in the background chapter, we chose WoFS as our source for NWP data as it is a high-quality modeling system for nowcasting problems. As such, it is a strong candidate for the discussed hybrid NWP/observation approach to predictors.

We included the same WoFS variables for the experiments that included the lightning observations and in the experiments that were exclusively using NWP data. For our models that included the lightning observations (discussed below), WoFS primarily supplied forecast environmental variables that were able to drive the evolution of the storms observed through the real-time observations. Despite this, some convective fields are still included so that at least some degree of correction away from the early observations can be made in the late stages of the forecast period. All model fields that are used from WoFS have been listed in Table 3.1. The role of WoFS data predictors in all our U-Nets is shown more explicitly in Figures 3.1 and 3.2 below.

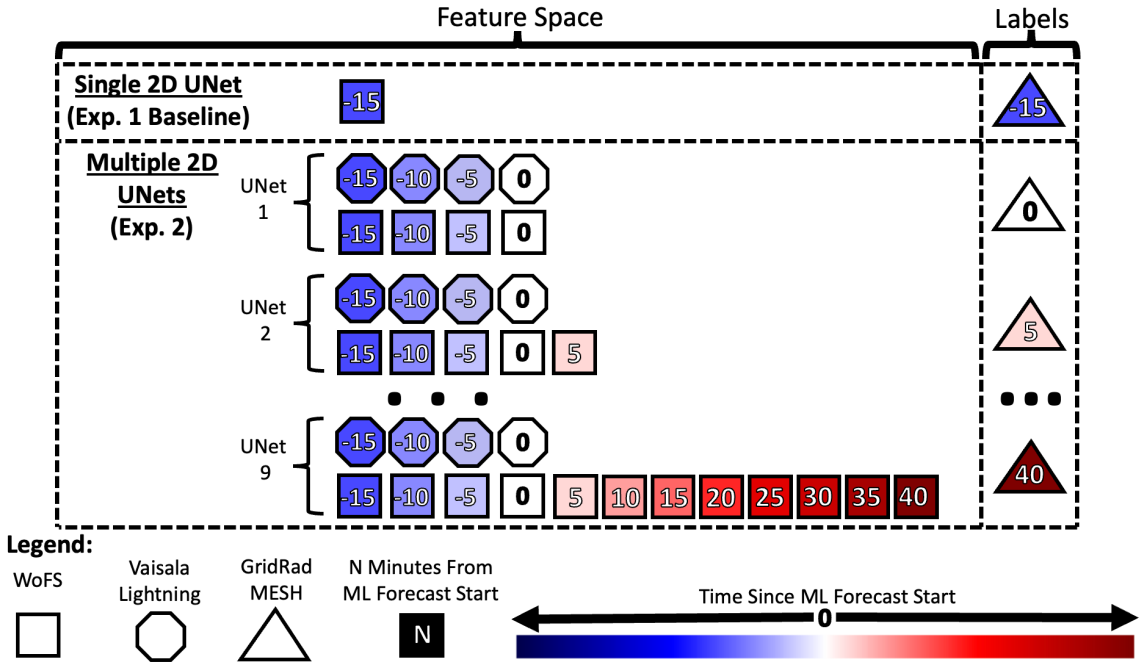



Figure 3.1: A diagram of the data slicing used in the first two architecture experiments. A legend is given at the bottom of the diagram. Each shape indicates a single timestep of the predictors. The colors indicate the time since the start of the machine learning forecast with the integers within each shape. Each dataset is indicated by a different type of shape.

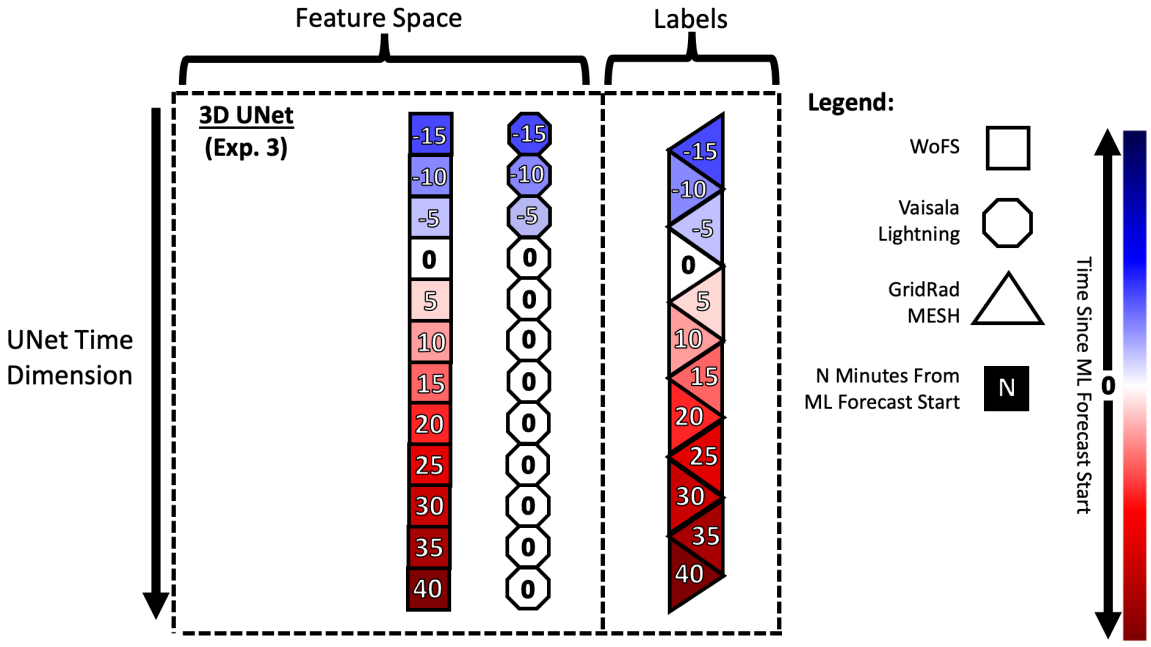



Figure 3.2: A diagram of the data slicing used in the third architecture experiment. A legend is given at the right of the diagram. Each shape indicates a single timestep of the predictors. The colors indicate the time since the start of the machine learning forecast with the integers within each shape. Each dataset is indicated by a different type of shape. The primary difference in this diagram when compared to figure 3.1, is that a third spatial dimension has been placed on the y-axis indicating that time is resolved in an extra dimension as opposed to feature space.

The fields we chose to use from WoFS can be broken into 5 categories: variables of convection, WoFS hail/severe weather composites, wind shear variables, humidity variables, and sounding heights. All derived fields were available from WoFS. Derived fields were chosen as opposed to raw fields because predictors that were more closely related to severe weather were expected to assist in the easier training of our machine learning models. Our variables of convection include: upward vertical motion (w-up), surface-based convective available potential energy (CAPE, which is the potential energy from buoyancy if an air parcel is able to rise through the atmosphere), most-unstable CAPE, surface-based convective inhibition (CIN), and most-unstable CIN. Our chosen severe weather composites are made up of: a standard Weather Research and Forecasting Model (WRF) graupel parameterization (labeled simply as "hail"), HAILCAST, and supercell composite parameter (Thompson et al. 2003). Our wind shear variables are: updraft helicity in the 2 km to 5 km layer, storm-relative helicity in the 0 km to 1 km layer, storm-relative helicity in the 0 km to 3 km layer, v-coordinate shear in the 0 km to 6 km layer, and u-coordinate shear in the 0 km to 6 km layer. Humidity was defined by dewpoint at the surface (2 meters). Finally, our chosen storm height values were freezing level, most-unstable level of free convection (LFC), and the most-unstable lifting condensation level (LCL). Again, these are all summarized in Table 3.1.

#### 3.1.1.2 Vaisala Lightning Detection

Although NWP has its numerous advantages, the short timescale between current observations and prediction time when performing nowcasting warrants the use of at least some observation data. Additionally, WoFS forecasts don't become available until $\sim$15 min after WoFS initialization (ML forecast start time) due to observation latency, the time required for data assimilation, model integration, and forecast output

post-processing. Observations should be able to help bridge this data gap in the early stages of our forecast. In particular, observations have been shown to add considerable skill to machine learning models used in the nowcasting timescale. Adding radar and satellite observations are popular sources for using observations (Czernecki et al. 2019). However, for this project it was decided that a data source directly associated with convection activity would be useful to use as a predictor because it was expected to more directly correlate with the position of convective features such as hail.

Vaisala's National Lightning Detection Network (NLDN) dataset was chosen for our observations because of its resolution and association with convection (Cummins et al. 2006, Orville 2008, Pessi et al. 2009). Additionally, it has an associated global dataset of similar quality for future upscaling. This dataset's use is based around a similar strategy to what was employed for our NWP dataset. An observational dataset with a specifically global domain was critical because making a global hail forecasting model is a planned future expansion to this system being developed.

Lightning events are derived from land-based NLDN observations in the Continental US before being stored as a collection of coordinates. What was used for this thesis was the number of lightning events summed together for each 3 km by 3 km pixel within our labels domain (see below section). A binning algorithm was used to detect the number of events within each pixel based on the coordinates of each event. Land-based observations have an advantage over satellite-based observations as they have less issues compensating for the curvature of the earth when compared to detection from aloft. This is expected to increase the spatial accuracy of the recorded lightning events. Specifically, Vaisala claims median location accuracy within 100 meters for their lightning data (Vaisala 2023).

For the purposes of this thesis, we chose to set observation time using WoFS initialization time as a reference. The same amount of observation data is used for each

WoFS forecast run regardless of how much real time has past since WoFS initialization. This decision was made because WoFS initializes every 30 minutes during severe weather events. Therefore, by the time a machine learning forecast with old observation predictors would start to become stale, a new WoFS run would have started and any new observations could be fed into the machine learning forecast created from this new run. However, this system would likely have to be rethought when scaling up this product to a global domain where NWP initialization would be less frequent.

For this thesis we selected 15 minutes of observations because this was the time it takes for WoFS spinup to complete (as discussed earlier). The position of this predictor in our U-Nets was described in more detail in Figures 3.1 and 3.2.

### 3.1.2 Labels (GridRad MESH)

In the field of using machine learning to forecast severe weather, storm reports are among the most popular sources for truth labels and performance verification (Lagerquist et al. 2017). This has several advantages including, but not limited to the confidence given by in-situ observations (or visible confirmation) of a particular severe weather event. However, this source of truth labels also has several disadvantages. One such disadvantage is that human-based measurements can have inconsistencies and biases. For example: the measurement of hail is often done in set fractions of inches and thus what constitutes rounding to these fractions can differ among people making observations (Allen and Tippett 2015, Blair et al. 2017). Another issue is that there has been shown to be a rural bias with storm reports (Cecil 2009, Potvin et al. 2019) that results from most reports originating near people's homes in urban centers as opposed to reporting out in sparsely populated areas. In contrast, a radar-based product would not have any introduced human bias, however it has some limitations inherent to radar systems. For example, the altitude of the radar beam increases with distance from the

radar site due to the curvature of the earth. This results in measurements of precipitation very high off the ground. This implies that the observed precipitation at the ground can be different to what the radar returns from the precipitation detected aloft. This also implies that the horizontal positioning of precipitation detected by the radar can be offset from where the precipitation lands on the ground. However, another positive attribute of the radar data is that it already exists in a grid-based format (as opposed to the sparse data of storm reports), which is useful since we are attempting to produce a grid-based product.

For this project, we elected to use the maximum expected size of hail (MESH) (Witt et al. 1998) calculated from the gridded NEXRAD WSR-88D radar (GridRad) dataset as our radar-based truth labels (Murillo and Homeyer 2019, Murillo et al. 2021, School of Meteorology/University of Oklahoma 2021). Specifically, this was sourced from the GridRad-severe distribution which used version 4.2 of the GridRad algorithm (School of Meteorology/University of Oklahoma 2021). It was decided that giving up the advantages of having mostly reliable in-situ observations in an effort to reduce biases was worth the exchange. Unfortunately the use of a hail surrogate such as MESH also has some disadvantages. MESH is known to struggle with hail at the smallest and largest scales (Cintineo et al. 2012, Ortega 2018). However, one of the reasons the GridRad dataset was chosen as opposed to other gridded radar datasets with MESH was that GridRad has updated MESH calculations that partially compensate for this (Wendt and Jirak 2021). This newer version of MESH was created based on a larger sample dataset than what was used in (Witt et al. 1998).

MESH was originally creating by fitting to the severe hail index (SHI) (Witt et al. 1998) via power law, which is a direct link between radar reflectivity and hail. In addition to using a higher quality data source, GridRad improved on this fit with its updated MESH (Murillo and Homeyer 2019). GridRad produced two separate power

law fits to SHI, one was to the 75th-percentile of the hail distribution ($MESH_{75}$) while the other was to the 95th-percentile of the distribution ($MESH_{95}$) (Murillo and Homeyer 2019, Wendt and Jirak 2021). It was found in development of GridRad that $MESH_{95}$ had an overall better performance increase over the original MESH (Wendt and Jirak 2021). However, they also found that $MESH_{75}$ performed better for hail around the 40 mm diameter, while $MESH_{95}$ performed better around the 64 mm diameter (Wendt and Jirak 2021). For this thesis the 95th-percentile option ($MESH_{95}$) was chosen because larger hail performance was deemed more important than smaller hail performance because we were approaching the problem from a no-severe-hail/severe-hail binary standpoint. Greater reliability on the severe-hail labels was deemed essential. Additionally, $MESH_{95}$ was found to be less cluttered (less labels produced), which was theorized to help keep the issue of model over-prediction (which was common in this thesis) under control.

Our best models were all built around the prediction of severe hail (hail at least 1-inch in diameter). To convert MESH data to labels that could be used in a binary segmentation problem, a threshold was needed to create a grid-based mask. When using 1-inch as the threshold, all pixels with at least 1-inch MESH derived hail were labeled as positive, while all other pixels were labeled as negative. For some experiments we tried 0.3937 inches (10 mm) to simulate general hail prediction (any size) which resulted in the same binary system. These thresholds would hopefully mitigate MESH issues, however it is most important that we acknowledge the limitations of MESH in this section as it is possible our own prediction's quality has a hard limit imposed by how well MESH itself performs for a given storm event.

### 3.1.3 Other Data (GridRad Reflectivity)

Finally, the last data source used for this project was the GridRad dataset (School of Meteorology/University of Oklahoma 2021) composite reflectivity field (as opposed to a more derived product such as MESH). This data source was not used as either a predictor or a label. It was included so it could be used in post-training statistics. Specifically, it was used to assist with determining how well WoFS predicted convection for particular days. This was deemed to be important as any failures of convection prediction in WoFS was found to be a significant source of poor performance for any model trained without the lightning observations. This was because at least some amount of convection within our WoFS predictors would be necessary for our models to predict hail when they did not have lightning observations. This could be understood as our models avoiding guessing the positioning of their own convection without either WoFS or lightning observation guidance. When lightning observations were included, earlier timesteps could use the observations as a marker for convection and could then advect forward predictions a few steps. In this case, the quality of WoFS convection was less critical, however it was still useful to use this reflectivity data as evidence of dataset variance while justifying the need for the data shuffling system we devised. More details on the use of this reflectivity data source in the context of data shuffling in our results chapter (chapter 4).

## 3.2 Data Mining and Pre-processing

The full process of data mining and pre-processing is described at a high-level by Figure 3.4. To prepare data for consumption in a U-Net, we need to slice gridded data into square segments with a width that is a factor of 2. A factor of 2 is needed since the down-sampling stage of our U-Net reduces the size of the square by 2 with each layer.

(See chapter 2 for more details on the workings of U-Nets.) From this point on we will call these squares "patches" to distinguish them from complete images that would be traditionally used to train a U-Net since these patches are segments of larger meteorological domains. While the patches had to be sized according to a power of 2, the particular power of 2 chosen could be varied as an experimental variable. Additionally, which WoFS fields were used in each U-Net, whether the lightning observations were used, and the distribution of hail all were experimental variables. For all of these options, multiple datasets with varying spatial and temporal domains must be combined in order to produce valid patches. Datasets with different projections and different resolutions would have be re-sampled to fit together with each other. Each dataset also had different formats for keeping track of their position in time.

### 3.2.1 Data Mining (AutoPatcher)

All these factors made it clear that a robust piece of software for producing patches from any combination of datasets would be required to avoid unnecessary work caused by hardcoding the patching process for each dataset. This software is made up of one master program that only required a set of configuration files to define a particular experiment. The result was the development of the AutoPatcher project, which has already proven robust enough to be used in other AI2ES U-Net-based projects. (For more details on this project, see Appendix A.2.) This was a system for generating patches from any given dataset that was controlled by a set of easily customizable configuration files. The end result of this system was a piece of software that could generate patches from any number of distinct datasets with differing data organization strategies and formats. This was important for a thesis which required predictors from many different sources. It also allowed for easy tuning of data-specific hyperparameters

throughout the project. This resulted in a large range of machine learning models generated and tested by the project's conclusion.

### 3.2.2 Preprocessing

Once the data had been selected, filtered, modified, and patched, each patch had to go through a preprocessing phase in preparation for consumption by the U-Nets. This preprocessing phase consisted of several steps which varied slightly depending on the U-Net architecture used in a particular experiment. All preprocessing pipelines are summarized in Figure 3.4. These architecture experiments were broken down into three categories:

**Architecture Experiment 1:** A single traditional two-dimensional U-Net with no time component that is trained at WoFS initialization time only with no observation sources. Used as a baseline.

**Architecture Experiment 2:** Multiple two-dimensional U-Nets with the time dimension converted to the feature dimension. Observations are included in the first 15 minutes of forecast time because these minutes are considered spinup time.

**Architecture Experiment 3:** A single three-dimensional U-Net where the time dimension was converted to the third spatial dimension.

These three categories are described in more detail in 3.3, however, in this section the focus is given to the data's role in these structures. For all three of these categories, the patched data is returned from our AutoPatcher in the same consistent format to ensure uniformity across experiments. Then, for each experiment, the data was sliced according to their respective predictor requirements. For example, the second

experiment was made up of several distinct U-Nets with differing sets of inputs, so slicing could be done to this raw data returned from the AutoPatcher for each individual U-Net without having to regenerate data. This enabled us to incorporate 15 minutes of observation data at the start of our forecast period in several different formats without extra oversight. In summary, each patch returned from AutoPatcher contained 15 minutes of lightning observation data, a full one-hour swath of WoFS data, and a corresponding full one-hour swath of MESH radar observations to be used as labels for each time step. Samples that were then needed for each architecture experiment were sliced from this full patch format to fit their corresponding needs in the early phases of preprocessing.

#### 3.2.2.1 Test Set Partition and Storm Day Clustering

Before data slicing for each architecture experiment, the test set had to be partitioned from the rest of our data with complete statistical independence and without any introduced selection bias. Since each patch was built around a one-hour segment of a WoFS run, many patches could be nearly adjacent to each other temporally or at least contained storms from the same storm event. Therefore, the traditional random selection of samples could not be used to partition the test data set in our application. To correctly partition the data, entire "storm days" would have to be the items that are randomly sampled to form the test data set. It was clear that "storm days" had to be formally and mathematically defined such that the sampling could be robustly performed.

It was decided that clustering the WoFS initialization times of like-patches together would be the most concise way to define a "storm day". This was a simple one-dimensional clustering task that could be solved by using the difference in time

between patches' WoFS initialization times. This could be described as a density-based clustering problem and thus DBSCAN (or Density-based spatial clustering of applications with noise) was selected as our clustering model (Birant and Kut 2007). We defined the time to be at least six hours between two patches' WoFS initialization times for both to be considered as part of separate "storm days". In DBSCAN terms this was defined as an epsilon of 6 hours. This resulted in a total of 68 "storm days" for our entire 2017-2021 data set. From this collection of patches now labelled by DBSCAN, 20% of all "storm days" were randomly selected to be set aside as our test data set where no action would be taken on this set until the end of our complete training process for all experiments. This method of clustering was also used in later steps of the preprocessing pipeline as it was important to define "storm days" whenever any sort of data partitioning was required. In particular, see 3.2.2.3 regarding cross-validation where this was used extensively.

#### 3.2.2.2 Minima/Maxima and Architecture Data Slicing

With the test set partitioned, the remaining data was defined as the joint train and validation sets. The minima and maxima of all variables contained within this joint set were calculated and set aside for later use in normalization (see section 3.2.2.4). Afterwards, the train and validation data was then ready for slicing into the respective structures required for each of the three U-Net architecture experiments (section 3.2.2). For architecture experiment 1, the data was sliced to only contain WoFS data and MESH labels at WoFS initialization time as discussed in sections 3.2.2 and 3.3. The single U-Net used in this architecture experiment was trained exclusively at a single timestep and used to predict on any of the other timesteps as a simple U-Net baseline. Therefore, no lightning observations were included in this architecture experiment as

it was possible that the U-Net would be used to predict on a timestep past valid observation times.

For architecture experiment 2, a considerably more complex slicing system was required. As described in sections 3.2.2 and 3.3, unlike the baseline, this architecture experiment did resolve the time dimension by converting each time step (with corresponding predictors) to the feature space of the U-Net. This "conversion to feature space" process was achieved by selecting each consecutive time step within the predictors and renaming them by their corresponding timestep. For example the WoFS feature "most-unstable CAPE" can be renamed as "most-unstable CAPE 0", "most-unstable CAPE 1", "most-unstable CAPE 2", and "most-unstable CAPE 3" for use as predictors in a two-dimensional U-Net predicting forecast minute 15. However, in this case the U-Net that formed the basis of this architecture experiment was still a two-dimensional U-Net and therefore would still produce two-dimensional probability distributions as output. Thus, to achieve a forecast for the entire one-hour period, multiple two-dimensional U-Nets were required in sequence to resolve the third time dimension. For each of these consecutive U-Nets, the forecasted minute was shifted 5 minutes later in the forecast period and an additional 5 minutes of predictors was added with the exception of lightning observations (since observations could not be observed later than 15 minutes into forecast period). This resulted in a total of 9 two-dimensional U-Nets (and 9 separate data slicings) for a complete forecast period from initialization + 15 minutes (ML forecast start time) to initialization + 55 minutes (ML forecast start + 40 mins). See Figure 3.1 for overview.

Finally, for architecture experiment 3, very little slicing was required. The entire forecast period's data swath could used in its original "extra time dimension" format (the lightning observation data's final 15 minute timestep was copied forward to fill the remainder of the time dimension for dimension shape consistency). All data was used

simultaneously since the time dimension was treated as the third spatial dimension needed for a three-dimensional U-Net (see 3.3). Slicing the combined train and validation dataset into the formats required for the three architecture experiments resulted in 11 total datasets for 11 total U-Nets (architecture experiment 2 required 9 individual U-Nets).

#### 3.2.2.3 Cross-Validation

Despite having 5 years worth of data (2017-2021) and 68 storm days, the rarity of hail events led to the conclusion that cross-validation was required to ensure over-fitting would not occur on the validation sets. Cross-validation achieves this goal by producing multiple independent validation sets from the original data set. For the purposes of this project it was determined that 5 folds (and therefore 5 independent validation sets) were sufficient to achieve confidence in validation results. Both 5 folds and 10 folds are popular choices for K-fold cross-validation (Allen 1974, Stone 1974), however it was determined that 10 folds would result in validation set sizes that were too small to be useful for this project. Additionally, the training of multiple complex U-Nets is exceedingly computationally expensive, thus multiplying compute time by a factor of 2 with the 10 fold option was deemed unreasonable.

In preparation for cross-validation, the 11 sliced datasets described in the previous section were then run through the DBSCAN-based storm clustering algorithm which itself is described in 3.2.2.1. However, at this stage the clustering algorithm labels patches based on their storm day for use in a stratified K-fold cross-validation partitioning procedure rather than random sampling. These labeled storm days were used for exploiting Stratified Group K-fold cross-validation (Stone 1974). Stratification is the process of ensuring each train and validation set partitioned with each cross-validation fold has hail label proportions (the number of hail labels divided by the total number

of labels) that are as similar as possible. This was again important for the rare hail event issue consistently encountered by this project. Additionally, this cross-validation function could be described as group cross-validation because it performed stratification within the constraints of predefined label groupings. Stratification with grouping worked by enforcing hail label proportion similarities without allowing two patches from the same group to exist in both training and validation. This grouping system was where the clustered storm day labels were fed. The result of this cross-validation stage was 5 splits of train and validation pairs for each of the 11 U-Nets for a total of 55 models requiring hyperparameter searches and training.

#### 3.2.2.4 Normalization and Remaining Preprocessing

As mentioned in the architecture data slicing section (3.2.2.2), the minima and maxima of all variables were calculated and set aside. These minima and maxima were kept to perform min-max normalization (equation 3.1). After all the original remaining non-test data had been sliced into the 55 train and validation pairs, min-max normalization was performed on all 55 based on the same minima and maxima. The result was U-Net predictors that would range from 0 to 1. This scaling matched the scaling of all our U-Nets' chosen activation function referred to as the Sigmoid function (equation 3.2). This had the advantage of instilling greater numerical stability during the model training process.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3.1}$$

$$S(x) = \frac{1}{1 + e^{-x}} \tag{3.2}$$

Immediately following the normalization of the predictors, all train and validation data was formatted to meet the requirements of a Tensorflow dataset (Abadi et al. 2015). A Tensorflow dataset was an efficient data storage method that consisted of both train and validation data merged together for consumption in a Tensorflow machine learning model (Abadi et al. 2015). The formatting included: required dataset reshaping according to Tensorflow standards and the conversion of all data to float32 for disk space consumption reasons. For certain experiments, a "Gaussian expansion" (Earnest et al. 2023) is applied to either the training set or both the training and validation sets simultaneously (see Figure 4.2 for results of this experiment). This "Gaussian expansion" step was implemented in response to the "rarity of hail events" issue discussed in the previous section. It worked by converting isolated hail labels (or 1's) to Gaussian-esque distributions of 1's surrounded by rings of 0.66's which were in turn surrounded by rings of 0.33's. This expansion was applied exclusively to the two spatial dimensions used for the 2 out of 3 architecture experiments made up of two-dimensional U-Nets. However, the expansion was applied to both space and time for architecture experiment 3 (which was made up of a three-dimensional U-Net). The objective of this was to allow for some amount of spatial and temporal tolerance on the prediction of hail in a gridded format as opposed to just exact pixelwise accuracy of prediction.

## 3.3 Machine Learning Architecture (U-Nets)

U-Nets were broadly introduced in the background chapter 2. Our application of these networks can be briefly summarized with a listing of the particular U-Net subtypes we used, along with their aforementioned associated predictor dataset slicings. As mentioned in 3.2.2, three separate U-Net architecture experiments were performed in

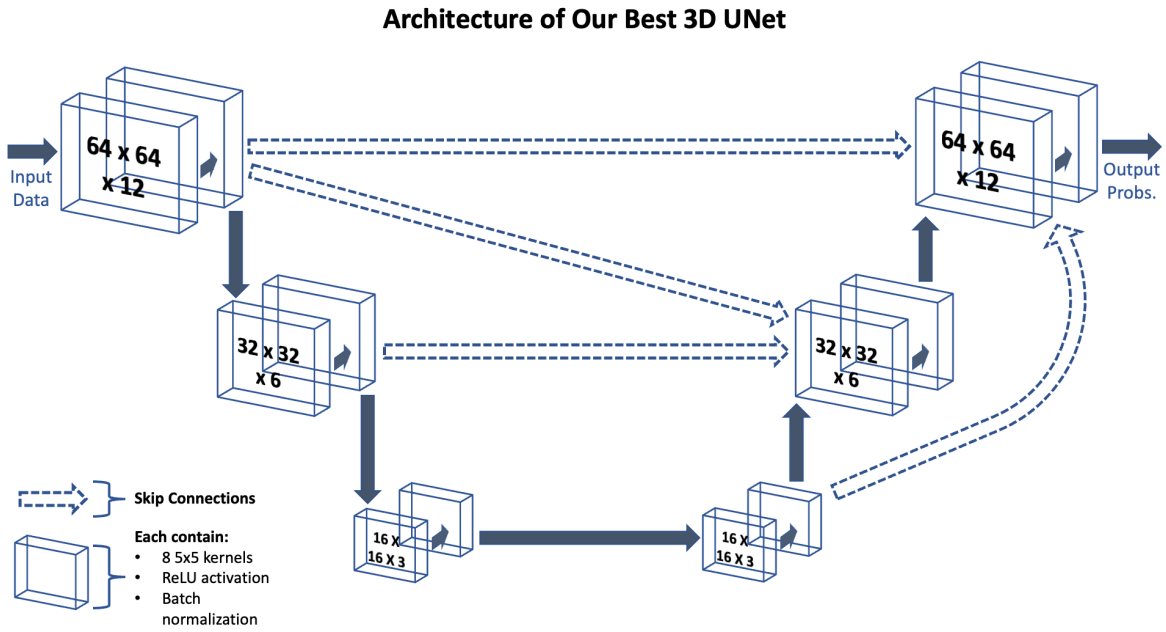



Figure 3.3: An overview diagram of the architecture for our best model. This overview is also a glimpse into what U-Nets look like in general. This model was a single 3D U-Net used in architecture experiment 3. All its convolutional layers are 3D and are displayed as such. This particular model had 2 convolutional layers per network level and skip-connections at each level because it is of the 3-plus variant. Each convolutional layer had 8 5x5 kernels, batch normalization, and a ReLU activation function. All hyperparameters for this model and all others produced are available in Table A.2

this project. The U-Net used for our best model is given in Figure 3.3. All the now preprocessed data was fed into the U-Nets used in one of these 3 experiments depending on which pipeline it was sliced for.

The first experiment used a simple two-dimensional U-Net trained at only one timestep (WoFS initialization time). It was designed to predict at any single timestep throughout the forecast period. As such, it did not get trained on any observations since observations were not available in all timesteps of the forecast period. This model was not expected to perform overly well and was included as a baseline for general machine learning performance.

The second architecture experiment used multiple two-dimensional U-Nets to each individually predict a timestep throughout the forecast period. Additionally, each of these U-Nets received predictors from all preceding timesteps as opposed to just using the predictors from the current timestep. This was done by moving all predictors from previous timesteps into the feature dimension of each U-Net. Since different timesteps could be resolved within these features, this experiment was capable of using the lightning observations.

The final experiment used a single large three-dimensional U-Net. This U-Net took the entire swatch of predictor data as input, as opposed to splitting it into chunks. The 12-step time dimension of all predictors were converted to the third spatial dimension. The lightning observations, which were only available for the first 15 minutes (the first 4 timesteps) of the forecast period, had its data at the 15 minute point copied forward the remaining number of times left in the forecast. This was so every predictor would still have the same shape when fit into the U-Net. This U-Net was our most successful and the same set of hyperparameters was used in all runs that used architecture experiment 3. This set of hyperparameters and the corresponding optimal architecture are displayed in Table A.2 and Figure 3.3 respectively.

## 3.4 Hyperparameter Searches and Model Training

At the conclusion of preprocessing we had 55 sets of data to train 55 U-Nets on. 45 of these U-Net and dataset combinations were made up of the 9 models necessary for architecture experiment 2. The remaining 10 combinations were made up of the 5 cross-validation folds needed for the other 2 architecture experiments that each only required a single U-Net.

With 55 individual U-Nets all requiring hyperparameter optimization, it would have been exceedingly impractical to perform costly hyperparameter searches for all U-Nets. As such, we elected to perform hyperparameter searches exclusively on the first fold of every cross-validation set for a total of 11 hyperparameter searches. The remaining folds were then only trained with optimal hyperparameters found by the best performing runs produced by the first fold searches. Our optimized models selected for final use and testing were then defined as models made up of hyperparameters that produced high performance results for all 5 folds. With this constraint, the computational cost of this procedure was within the bounds of resource limits available at AI2ES, while still enabling the production of a robust and reliable machine learning model system.

## 3.5 Max Critical Success Index Metric

$$CSI = \frac{TP}{TP + FN + FP} \tag{3.3}$$

We chose two metrics to represent the performance our our models. The first was a pixelwise maximum critical success index (max CSI). Critical success index was calculated from the base metrics as shown by equation 3.3, where "TP" is the true positive rate, "FN" is the false negative rate, and "FP" is the false positive rate. Max

CSI was where the maximum critical success index was taken across a given range of probability thresholds for the model output. This max CSI is the point closest to the top right along the curve of a performance diagram. The advantage of this method was that it allowed for variance in probability magnitude across experiments which is important for rare label cases. The second metric was max CSI with a 6-km neighborhood tolerance. This was calculated in the same manner as pixelwise max CSI except that the true positive and the false positive sums were calculated on truth labels that were expanded by 6 km. The true negative and false negative were still calculated on unexpanded labels. The result of this was a metric with a spatial tolerance that did not introduce additional unphysical false negatives (which would occur if one applied a label expansion to all four base metrics haphazardly). However, this neighborhooding did not include any temporal tolerance.

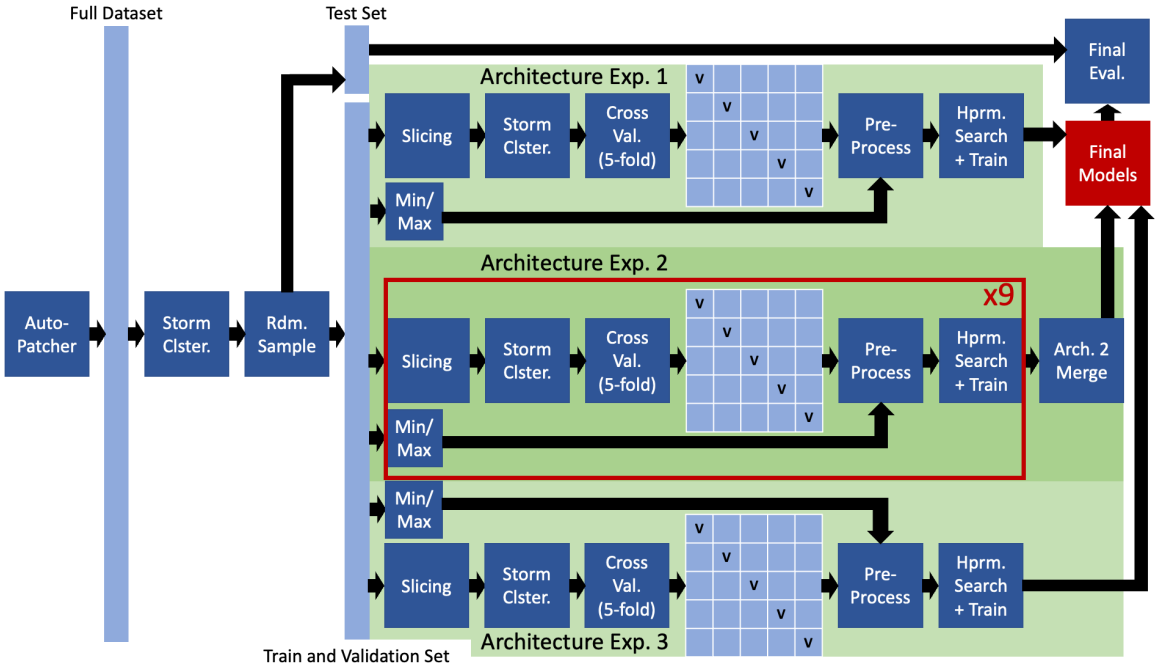



Figure 3.4: A flowchart of the entire data mining and preprocessing pipeline. Each dark blue square is a software step. Each light blue rectangle or square is data. Data shapes are not to scale. The green shades help delineate the 3 architecture experiments which are each labeled. All 3 architecture experiments have similar high-level structure, however architecture experiment 2 has to repeat its steps 9 times (as indicated by the red boxed section) to resolve all timesteps in our forecast period. Most steps in this diagram are discussed throughout chapter 3. The left most Auto-Patcher code box is additionally described in Figure A.1, while each of the slicing code blocks are additionally described in Figures 3.1 and 3.2.

# Chapter 4

# Results

## 4.1 Critical Importance of Data Distribution

An important discovery made during the course of our early experiments was that the ability of WoFS to accurately predict convection played a substantial role in model performance for models that did not use observation predictors. In the early stages of this thesis, the lightning observations that would later become critical to our best models were not yet available. As a result, all machine learning models we produced had to rely on exclusively WoFS data for convection information. Our train, validation, and test sets were also simply delineated by dates as opposed to the clustering system used in later stages.

As these dates were shifted during various experiments, it was discovered that the test set performance across experiments varied to a large degree. (These experiments are outlined in more detail in appendix A.1). It was hypothesized that this was a result of variations in WoFS convection performance across dates. Thus, we examined WoFS convection performance in the same context as the rare hail label issue. It was hypothesized that the rarity of storm events (rare hail label issue) and the corresponding limited dataset size (final experiments contained a subset of 68 storm days from 2017 to 2021) exacerbated the issues posed by varying WoFS convection performance. From examining case study results throughout the duration of the thesis, it was clear

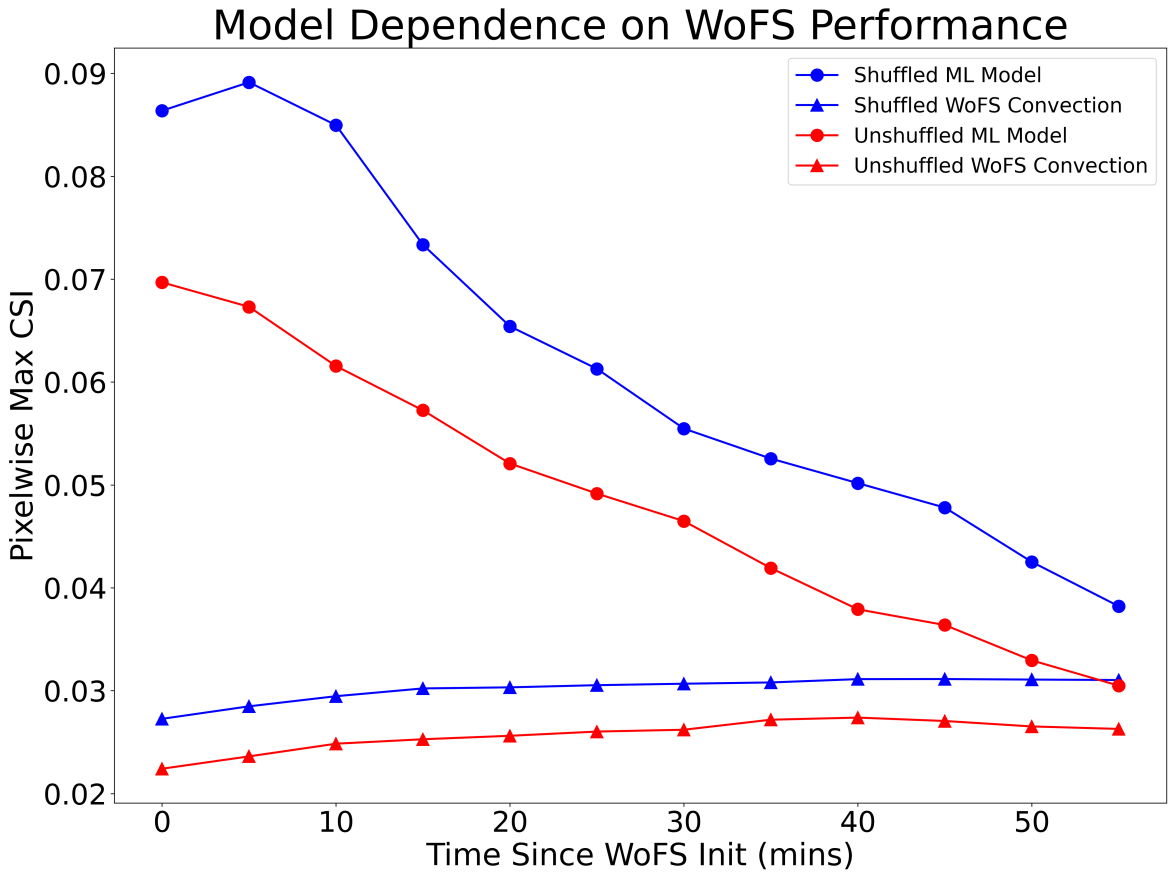



Figure 4.1: Results of an experiment indicating the relationship between non-observation-containing machine learning models and NWP performance. All points are pixelwise max CSI. The CSIs are a function of time since WoFS initialization. The red lines indicate test datasets that are unshuffled across time while the blue lines indicate test datasets that are shuffled completely with the storm clustering system. The lines with triangle markers indicate a surrogate for WoFS convection performance (CSIs from comparison of WoFS composite reflectivity >40 dBZ against GridRad composite reflectivity >40 dBZ) while the lines with circle markers represent the corresponding machine learning performance.

that WoFS convection performance varied substantially amongst samples and between ensemble members. Thus, it was also theorized that any unshuffled distribution of data could result in the division of unequal degrees of WoFS convection performance when again considering the context of a limited dataset size made up of relatively rare storm events.

To test these theories, an experiment was performed to compare a model trained on unshuffled data against one trained on shuffled data. The unshuffled model was trained on cross-validated data from 2017-01-01 to 2021-04-26 and tested on data from 2021-04-27 to 2021-12-31. The shuffled model used train, validation, and testing data shuffled according to our clustering algorithm over all available years (2017 to 2021). For a summary of dates see appendix A.1. For each model, patches were also created from the GridRad reflectivity dataset discussed in section 3.1.3 and set aside. The performance of both trained models were then be compared along with their respective WoFS convection performances. The convection performance was measured by producing the max CSI of WoFS composite reflectivity >40 dBZ with the set aside GridRad composite reflectivity >40 dBZ as the ground truth. 40 dBZ was chosen because reflectivity greater than this threshold usually was associated with convection and not stratiform precipitation.

The results of this experiment are outlined in Figure 4.1. The convection performance max CSIs are shown with the triangle markers while the machine learning model performance is shown with the circle markers. This experiment validated both hypotheses. Firstly, it is clear that any machine learning model produced had it's performance dependant on the WoFS convection performance. This can be seen from the fact the shuffled model's performance increased from the unshuffled model's performance at the same time the convection performance increased at a similar scale. Secondly, it is clear that shuffling plays a substantial role in increasing this convection

performance for the gathered test dataset. The results of this experiment indicate that any future machine learning endeavour that uses NWP data would need to investigate the performance of said NWP data while also considering if shuffling would be required. In order to maximize the performance of our model produced in this thesis, we used the shuffled method for our final model.

## 4.2 Predictor and Label Experiments

We performed 3 end-of-study predictor and label experiments. Each experiment was performed by changing a single condition from our most developed hail nowcasting machine learning model. This ideal model used architecture experiment 3 (see Figure 3.2 and 3.3) and employed the Gaussian expansion we discussed in 3.2 along with the Vaisala lightning observations discussed in 3.1.1.2.

The results of all 3 predictor and label experiments are summarized in line graphs given in each experiment subsection (see Figure 4.2 as an example). Each of these line graphs describes the complete test set performance of a model used in an experiment across the 1 hour forecast period with 21438 testing samples. Each figure included a line graph with pixelwise max CSI on the left and neighborhooded max CSI on the right. For all 3 experiments, each line graph was made up of the lines representing both models used in the experiment's comparison along with 3 baselines. In all experiments, each result was broken into output that exploited the ensemble distribution (by using the ensemble mean) of the WoFS data and output that was purely deterministic. The deterministic output is denoted by the models with circle markers in each figure. The ensemble mean version of each model is denoted by a triangle markers. All models were trained deterministically where all WoFS ensemble members were treated as reasonably equivalent and thus were merged together into a larger sample collection (a "NWP

ensemble member agnostic" approach). This was important for adding additional samples to counter the rare label issue. Thus, to produce ensemble probabilistic output, the models trained on the whole expanded dataset were used for prediction on test data segregated by ensemble members before we took the ensemble mean. For this thesis, it was deemed important to explore machine learning models that used both numerical weather prediction ensembles and deterministic numerical weather prediction models as predictors. It was decided to explore both because although model ensembles are most common, IBM's global GRAF model is exclusively deterministic. Since our long-term goals include applying this work to the GRAF output, we wanted to train a machine learning model on data that is as similar as possible.

The three aforementioned baselines are made up of WoFS ensemble-based probabilistic HAILCAST, deterministic updraft helicity-based logistic regression, and ensemble mean updraft helicity-based logistic regression. HAILCAST is the primary physics-based hail product used by WoFS. It is deterministically produced by each WoFS ensemble member. Therefore, the ensemble distribution of this product is required to make it a probabilistic variable that can be meaningfully compared to the probabilistic output of our models. Thus, this is the only variable that does not have any deterministic lines. The remaining two baselines were both variations of updraft helicity-based logistic regression which is a relatively simplistic machine learning model. These were included to assert the baseline performance of machine learning models. It was also deemed useful to include these two baselines as they are far less computationally expensive to run operationally relative to deep learning such as U-Nets. It was important to beat these two baselines with our machine learning models to assert their computational viability. The ensemble mean version of this updraft helicity-based logistic regression baseline was created using the same method as described for our machine learning models.

### 4.2.1 Gaussian Expansion Experiment

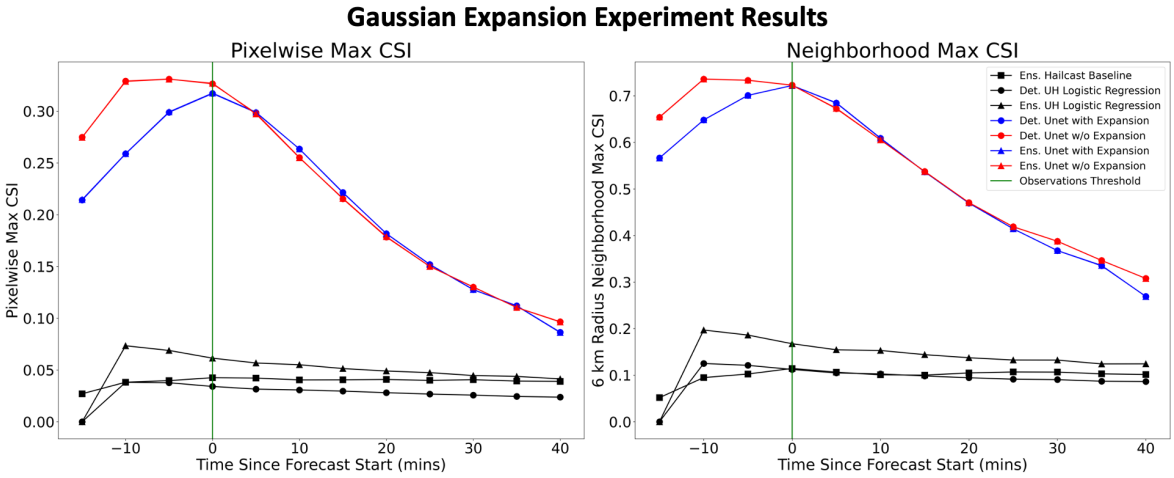



Figure 4.2: The results of our first predictor/label experiment. This experiment removed/added the Gaussian-based label expansion used in our train and validation sets. The left plot displays results using a pixelwise max CSI while the right plot uses max CSI with a 6 km radius neighborhood. Both plots examine the CSIs as a function of time since the ML forecast start. The black lines are the baseline cases, the red lines are the non-Gaussian models, and the blue lines are the original Gaussian models.

The first experiment's objective was to investigate the effect of a spatial and temporal expansion of labels in the train and validation sets. It was hypothesized that the expansion of these labels would allow for an easier training process for our machine learning models as a label expansion was expected to artificially increase spatial and temporal tolerance of hail predictions. It also was hypothesized to improve performance as increasing the number of labelled pixels helps to fight the rare hail label issue. We used a "Gaussian expansion" to accomplish this task. This expansion was applied spatially for every model that used this process and also temporally for all time-included 3D U-Nets (architecture experiment 3). This expansion was applied to both the training set labels and the validation set labels for all models that used this process. To maintain the integrity of the test set, no expansion was ever applied to the test set labels.

The results of this experiment are summarized in Figure 4.2. The blue lines present in both plots represent our models that were trained with labels expanded by this method. The red lines represent our models that were trained without expanded labels. As introduced previously, all black lines represent various baselines. The line with square markers represents ensemble HAILCAST. The lines with circle markers and triangle markers represent deterministic and ensemble updraft helicity-based logistic regression respectively.

Unexpectedly, it was discovered that this expansion did not have a substantial effect on model performance. As shown in the figure, the models that included the Gaussian expansion did not have a max CSI that was substantially greater than models without the expansion. Conversely, applying a Guassian expansion to our model decreased model performance in the first timesteps of the forecast period during the time observations were included in the predictors. It was theorized that this early reduced performance was because increasing label counts at a timestep where observations are included tends to add unphysical labels to pixels where no corresponding observations would be present. At timesteps where observations that nearly represent truth are present, expanding labels can only increase the error.

Beyond the timesteps of the first 15 minutes, it was hypothesized that this experiment did not yield substantial results because the limited number of labels constricted the advantages gained by expanding each label. With such a small number of severe hail labels available in the original dataset, any sort of physically reasonable label expansion was not expected to increase the total label count by a substantial number. The goal of this experiment was to offer some tolerances to the machine learning models during the training process, however, these tolerances cannot be as impactful as desired without a larger number of original severe hail labels to have learning tolerances on.

### 4.2.2 Predicted Hail Size Experiment

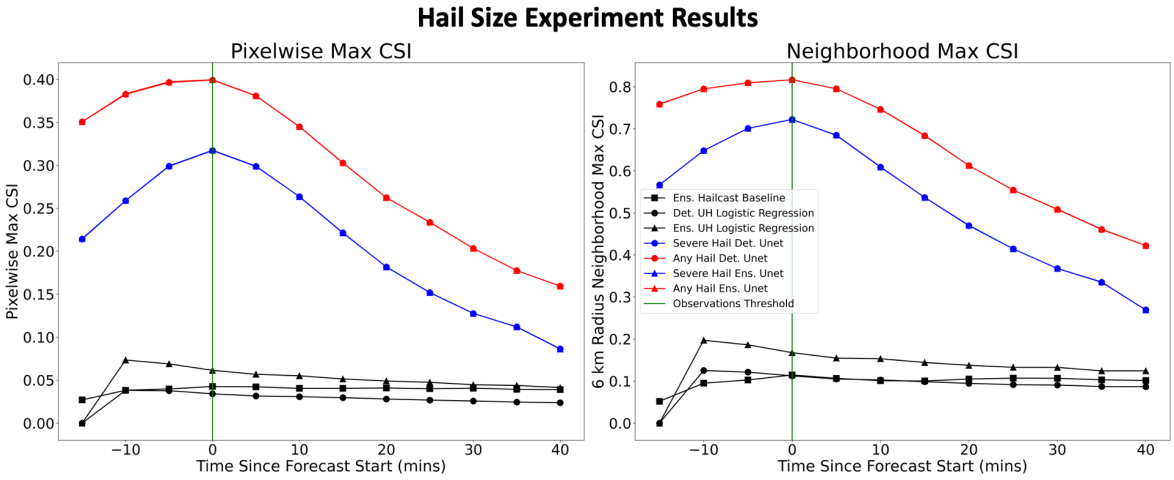



Figure 4.3: The results of our second experiment. This experiment examined the effects of changing our predicted hail size from 1 inch to 0.3937 inches (10 mm). The left plot displays results using a pixelwise max CSI while the right plot uses max CSI with a 6 km radius neighborhood. Both plots examine the CSIs as a function of time since the ML forecast start. The black lines are the baseline cases, the red lines are the any-hail models, and the blue lines are the original severe hail models.

The second experiment was performed to investigate the effect of the forecasted hail diameter on our machine learning model performance. This experiment was of particular importance as there is considerable motivation to forecast multiple sizes of hail in an operational setting. This smaller diameter forecast produced as a result of this experiment could also be perceived as an additional baseline comparison against our more impactful severe hail forecasts. This served to increase confidence in the validity and performance of our severe hail models.

This experiment is summarized in Figure 4.3. Once again the red lines represent the changed models (any-hail) while the blue line continues to represent our ideal models (severe hail). We referred to the smaller diameter hail size as "any-hail". We chose 10 mm as the threshold for "any-hail" because choosing 0 was considered too unstable

for useful machine learning predictions due to the GridRad MESH label performance dropoff.

This experiment's results were closer to what was hypothesized. Both the pixelwise and neighborhooded max CSI performance were substantially greater for the any-hail forecast across all timesteps. In the pixelwise plot, an increase as high as 0.15 CSI was observed for some timesteps. Since the any-hail forecast predicted a far more common event, it was expected that the model would perform better. This can be understood from the perspective of the relationship of one of our machine learning models with its label frequency. In the any-hail case, far more hail labels would be present which would increase the likelihood of labels overlapping with a dominant predictor such as WoFS composite reflectivity. This can be visualized as converting the forecasting problem to something more akin to a general convection forecast which should be easier for the any-hail models.

Overall, this experiment helped to emphasize the value of using multiple machine learning models with a basic binary output when producing a solution to a multi-task learning problem. This approach produced strong performing forecasts for two exceedingly different hail diameters while maintaining meaningful probability scales.

### 4.2.3 Lightning Observations Ablation Experiment

The final predictor/label experiment was an ablation experiment performed to investigate the effect of the inclusion of real-time observations in our predictors on our machine learning model performance. These observations consisted of lightning "event" data from the Vaisala NLDN dataset. The details of this dataset are discussed more rigorously in section 3.1.1.2. As discussed in section 3.2, these observations were only included in the first fifteen minutes of the forecast period for all models that included

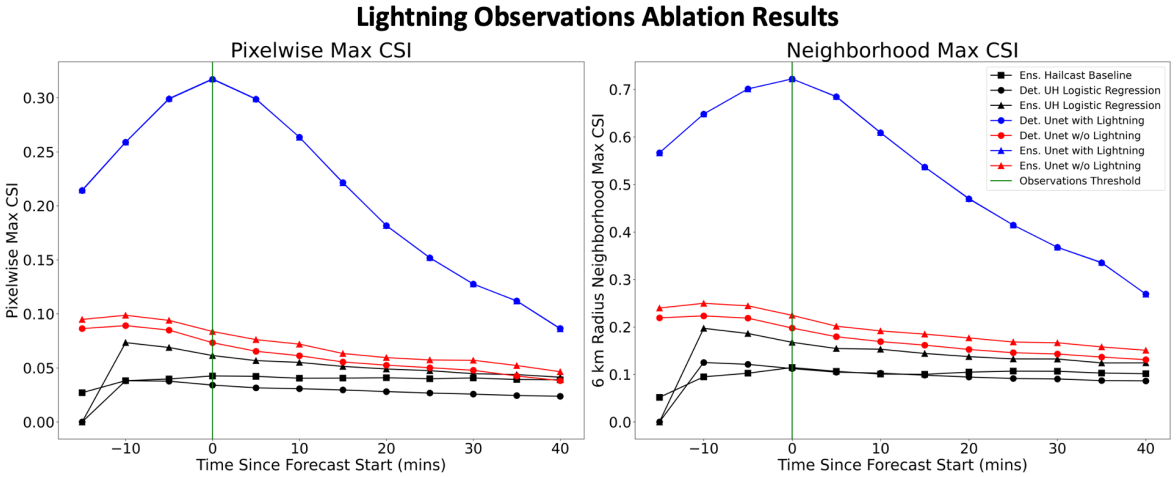



Figure 4.4: The results of our final ablation experiment. This experiment removed the lightning observations from the model's predictors. The left plot displays results using a pixelwise max CSI while the right plot uses max CSI with a 6 km radius neighborhood. Both plots examine the CSIs as a function of time since ML forecast start. The black lines are the baseline cases, the red lines are the ablation models, and the blue lines are the original models.

these observations. In all of our max CSI plots this threshold is denoted by the vertical green line. We considered this portion of the forecast period to have preceded the time at which our model becomes available for use in an operational setting. This green line could be considered the "present time" for a forecaster that was using this product operationally. Therefore, the most substantial timesteps occur after this line.

This ablation experiment yielded the most substantial results out of the 3 experiments. The models that contained lightning predictors are denoted with blue lines, while those without lightning predictors are denoted by red lines. As with the other experiment figures, deterministic models are denoted with circle markers, while ensemble-based models are denoted with triangle. The models that had their lightning predictors removed performed substantially worse at all timesteps. However, they still outperformed the baselines despite having no observation predictors, which indicates the general quality of this U-Net-based machine learning method.

In addition to the clear performance difference between the models that included lightning and those that did not, a striking difference in the sloping of the max CSI plots between the models was also observed. The largest difference in performance occurs in the first half of the forecast period, which aligns with the segment at which the observations are included. The models that contained lightning observations had a far greater difference in max CSI between the observations threshold and the final timestep when compared to the models that did not contain lightning observations. This result implied the value of observations to a machine learning model dropped rapidly as the forecast period advanced past the time at which the observations occurred.

Another result that was observed from this ablation experiment was the direction of the slope in the timesteps before the observations threshold. One hypothesized explanation for this unexpected sloping was based upon numerical weather prediction spinup times. It is possible that NWP spinup resulted in a poorer WoFS performance in the first timesteps of the forecast period. As WoFS would complete spinup and perform any needed initialization, an uptick in our machine learning model performance would be observed in the first 10 minutes of the forecast period as WoFS predictor quality improved. This would explain the smaller slope produced by the non-lightning models and part of the greater slope produced by the lightning-containing models. A second explanation for the greater slope produced by the lightning-containing models could be found in the lightning predictors' relationship with the WoFS predictors in the early stages of the forecast period. It is hypothesized that the lightning predictors' role in our machine learning model is partly to add confidence in the forecast in the early stages of the forecast period. This is because lightning activity alone cannot exclusively determine the presence of severe hail (Allen and Tippett 2015). A storm must first be correctly initialized and placed within an accurate WoFS environment for the lightning observations to offer meaningful contributions to the forecasting of a hail threat.

A further possible explanation for the slope at the beginning of the forecast period is due to a known "tapering-off" issue with 3D U-Nets (Bansal et al. 2022). All U-Nets have some degree of output quality dropoff near the spatial edges of the images they are segmenting. This is due to the translated kernels hitting the edge of the image in each step of the U-Net. A kernel that is unable to observe data outside of the image cannot produce meaningful convolved results on the edges. Since we have converted the time-component of our forecast to the third spatial dimension of U-Nets for all non-architecture experiments, this same kernel-based tapering-off limitation also occurred for our time dimension. This may possibly have resulted in our models producing less confident hail forecasts at the start and end of our forecast period. However, the max CSI used in all experiment figures is designed to supply a performance metric that is agnostic to the probability scalings of each timestep. Therefore, any reduction in absolute hail probabilities should be mostly ignored by our metric used in the plots. Further study into these competing effects is warranted.

The final result in this ablation experiment can be seen from close examination of the blue lightning model line. Both the ensemble-based and deterministic versions of this model are included in this figure just as what is visible for the non-lightning case. However, this is not immediately evident as the blue ensemble line is hidden behind the blue deterministic line. This indicates that there is no improvement to forecast performance for a model that uses lightning data when applying the model to the WoFS ensemble. A possible explanation for this is that our models that used the lightning observations switched the predictor used to drive their storm initialization from WoFS composite reflectivity to the lightning data itself. We believe this potentially occurred because WoFS environmental variables do not differ substantially across ensemble members, so an ensemble average using these variables would not alter our

model's performance. Therefore, the lightning model must only use these WoFS environmental variables as predictors while ignoring the more unstable WoFS convection predictors. To verify this hypothesis, any further studies around this model should include a feature importance investigation.

## 4.3 Architecture Experiments

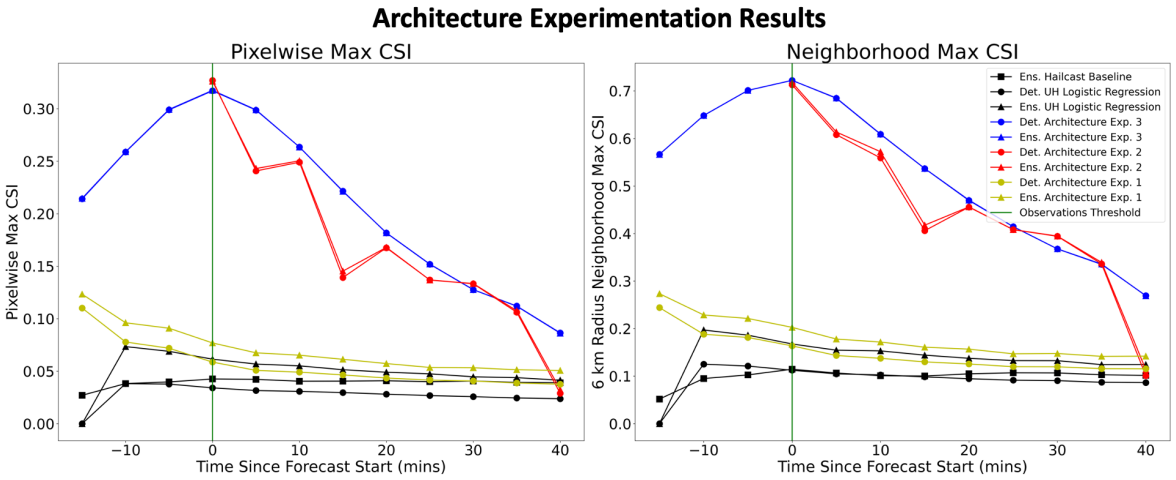



Figure 4.5: The results of comparing our three architecture experiments. The left plot displays results using a pixelwise max CSI while the right plot uses max CSI with a 6 km radius neighborhood. Both plots examine the CSIs as a function of time since the ML forecast start. The black lines are the usual baseline cases, the yellow lines are the 2D U-Net baseline models (architecture experiment 1), the red lines are from the cluster of time-resolving 2D U-Nets (architecture experiment 2), and the blue lines are the best performing models produced (the 3D U-Nets of architecture experiment 3).

The final test set evaluation we performed was a comparison of the architecture experiments (see section 3.2 and Figures 3.1 and 3.2). See Figure 4.5 for a summary of the results. Due to the evidence provided by our non-architecture experiments, which indicated predictor quality, lightning data were both used in all cases except the simple single 2D U-Net baseline (architecture experiment 1). Lightning data was excluded from this case because this particular 2D U-Net had no perception of time

and therefore had to independently predict output at each forecast timestep with each corresponding timestep's predictors. Thus, it was impossible to train this particular U-Net with lightning predictors as valid lightning predictors are not present in all timesteps. However, a Gaussian expansion was still used during training as this method could be applied to any timestep and applying it for this architecture experiment created at least some additional similarities to the other two architecture experiments.

Instead of representing different predictor and label experiments, each line in this section's max CSI plots represented a different architecture experiment. In this figure, the top overlapping blue lines continue to represent our best performing model. This model was made using architecture experiment 3. They are made up of the same architecture and feature set as was used in our predictor and label experiments. The red lines represent the second architecture experiment, where each of the two lines are made up of a temporal sequence of two-dimensional U-Nets. Each timestep for this architecture experiment is predicted by its own independent U-Net that uses the preceding timesteps' data as its predictors. Since each timestep required its own model and there is no advantage to forecasting in the observation time, only timesteps past the 15 minute threshold (00 on the plot) were forecasted. The final set of yellow lines represent our baseline architecture experiment 1, which is a simple two-dimensional U-Net trained exclusively on WoFS data at WoFS initialization time. This baseline model does not use the lightning observations as a predictor because these observations are not available throughout the forecast period and unlike architecture experiment 2 U-Nets, the baseline model only uses predictors available at the current timestep.

Architecture experiment 3 performed the best at nearly all timesteps, with 2 notable exceptions being the ML forecast start time and the 30 minute mark. This was the expected behavior, as its three-dimensional U-Net is capable of resolving structure in its third spatial dimension to the same degree as its other two dimensions. The smoothed

effect of the architecture experiment 3 max CSIs measured at all timesteps highlight this feature. We hypothesize that this smoothing implies that the previous timesteps are directly factored into the forecast considerations made in the succeeding timesteps. It can be described that the model knows not to vastly change the scale of its predicted probability of severe hail from the scale predicted in the previous timestep. The U-Net produces the temporal smoothing in the same manner it produces a spatial smoothing in traditional spatial segmentation tasks. Additionally, this has numerous advantages from the perspective of an operational setting. One such example is when any forecast is produced by this system, a meteorologist can have a greater degree of confidence in the physicality of the probability densities from timestep to timestep.

The results from architecture experiment 2 were more mixed. This particular architecture experiment was selected to contrast with experiment 3 because it offered a meaningful trade-off of advantages. In terms of total used predictors, this experiment used all of the same data as was used in experiment 3. However, it took on the disadvantage of losing the aforementioned smoothing effect produced when using a single three-dimensional U-Net. Each two-dimensional U-Net used in each timestep did not possess an extra spatial dimension that could be used to directly perceive temporal evolution. Instead, the time dimension of each predictor was expanded into copies of the predictors in each U-Net's feature dimension (see Figure 3.1) (for a total of n*t features where n is the original feature count and t is time). This resulted in a more jagged appearance to the architecture 2 max CSIs as the model trained at each time step produced forecasts with no perception of the forecasts produced at the previous timesteps. Each model was optimized in a different way to each other, resulting in varying max CSIs across the forecast period. The advantage that was gained from using architecture experiment 2 was a greater ease of training for each U-Net. Throughout

this thesis, it was often found difficult to train and optimize the more complex three-dimensional U-Net system, but a forecast would always be produced using architecture experiment 2. This architecture is also common in other nowcasting studies (or at least the kind of setup used in a single timestep of this architecture), so it was further useful as a comparison against our ideal experiment 3.

The final baseline single-timestep two-dimensional U-Net performed the worst as expected. It also performed worse than our 3D U-Net without lightning observations that was represented by the red lines in the figure used for our ablation experiment (Figure 4.4). This made sense as this architecture experiment also did not have any lightning observations while also being a more limited U-Net with no temporal resolution at all. One interesting result from this baseline experiment was that the "tapering-off" issue due to model spinup that is usually present at the start of each forecast was not observed. This is because this particular U-Net was trained at exclusively WoFS initialization time and therefore was prone to predict better on the erroneous model data at timestep -15 while rapidly declining in quality in succeeding timesteps.

## 4.4 Overall Performance

High-level results can be derived from all predictor/label experiment results and architecture experiment results. It was clear that all U-Net-based predictor/label and architecture experiments performed better than all base-lines (with a partial exception of the simple U-Net baseline). When the lightning observations were removed in its ablation experiment, CSIs dropped by 0.1-0.2 across most timesteps in the forecast period. However, despite this disadvantage, the deterministic U-Net without observations still outperformed the best deterministic baseline, while the ensemble-based

U-Net without observations outperformed the best ensemble baseline. In summary, all primary models we produced outperformed all baselines.

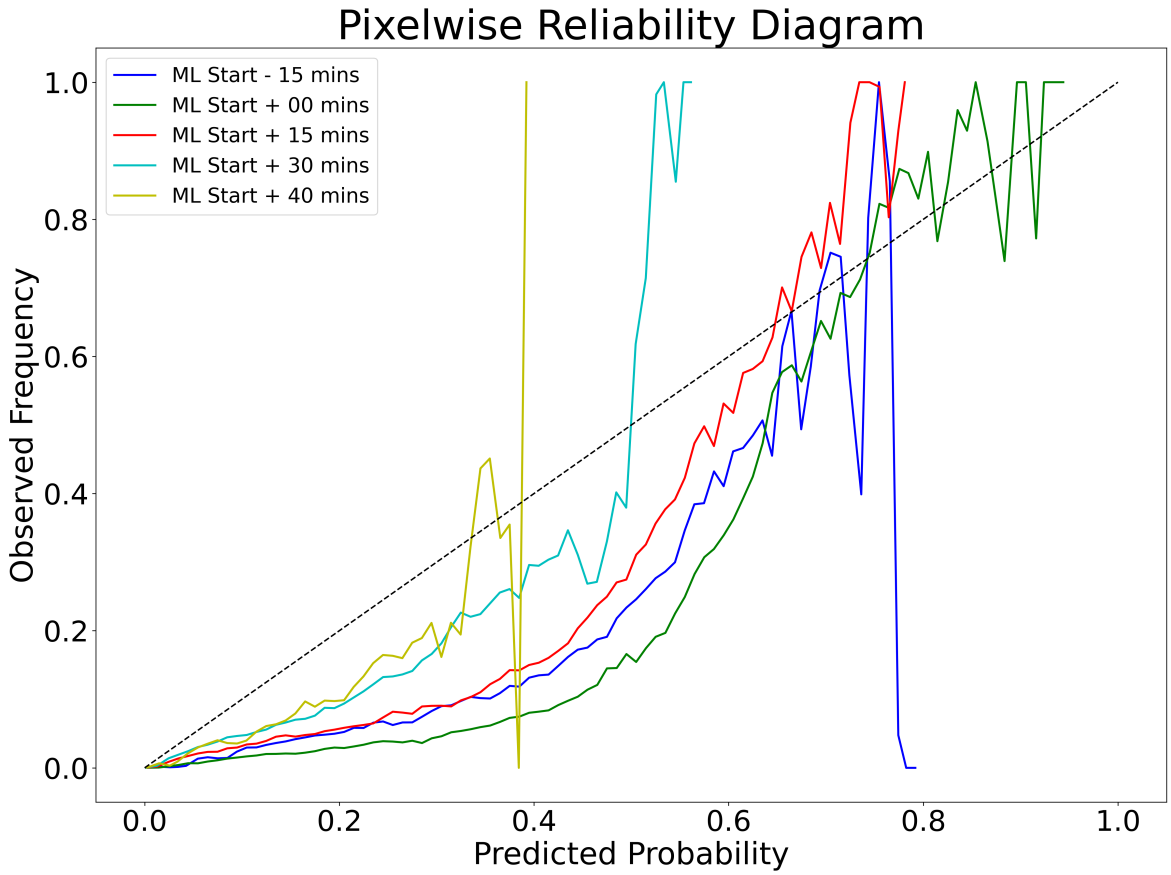



Figure 4.6: A diagram indicating pixelwise reliability for our best model at 15 minute intervals of the forecast period. The center dashed line indicates ideal performance.

Our best performing architecture and predictor combination by a large margin was the 3D U-Net (architecture experiment 3) with lightning observations included. Our pixelwise max CSI figures indicated max CSI values ranging from ∼0.32 at the timestep 00 to ∼0.10 at the final timestep. ∼0.32 is several times the magnitude of the max CSI values of the baselines at the first valid forecast timestep, while ∼0.10 is approximately double the performance of the baselines at the final timestep. This result exceeded our expectations and instilled confidence in the quality of the 3D U-Net's forecast.

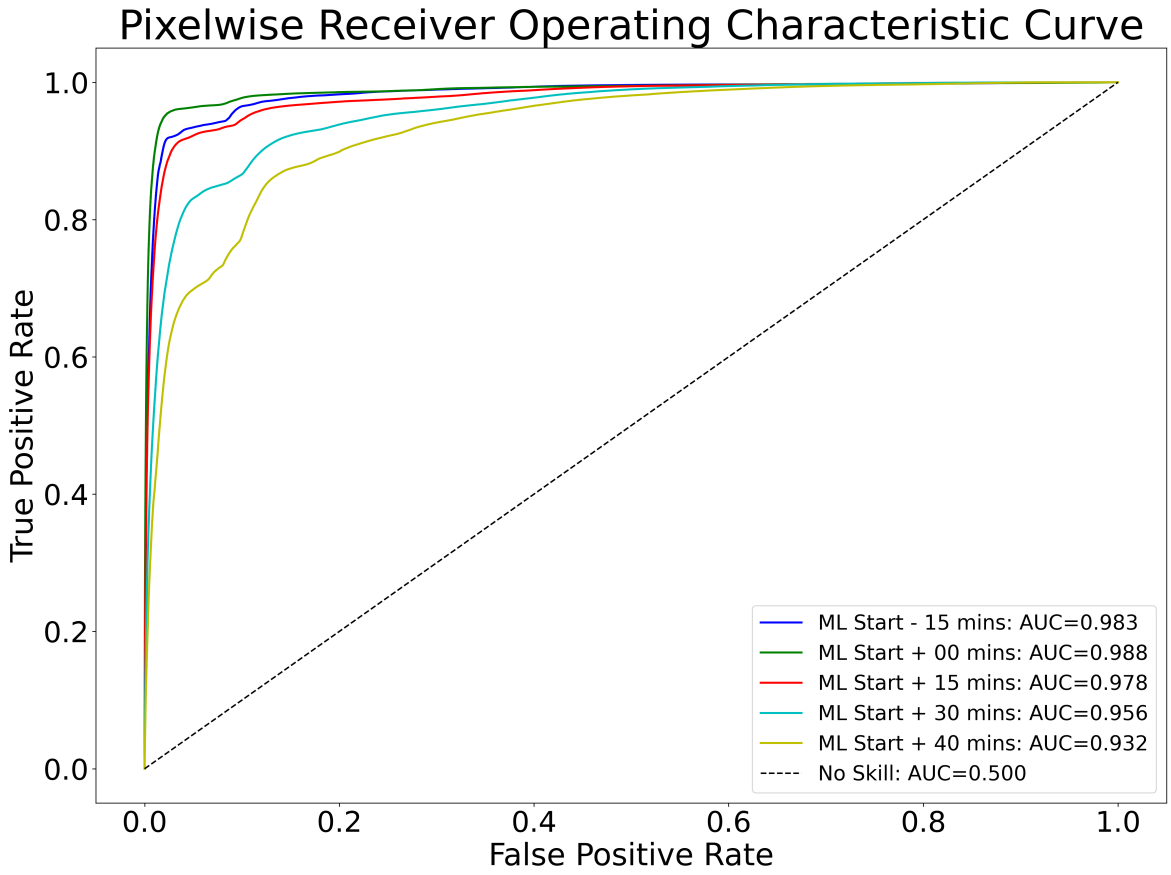



Figure 4.7: A diagram indicating model performance using area under the curve for our best model at 15 minute intervals of the forecast period. The closer a line is to the top left corner of the plot, the better its model performance.

A further result from the overall analysis of our best performing architecture across all max CSI figures was found in the neighborhooded max CSI plots. These plots showed that the 3D U-Net 6 km neighborhooded max CSIs had an improvement relative to the baselines ranging from ∼0.5 to ∼0.25 in the first half of the forecast period (from 00 to about 20 minutes past the ML start time). This made physical sense because the lightning observations would verify any ongoing convection at the start of this sub-period and therefore any hail prediction made during these timesteps would be spatially close to any real hail. The 6 km radius for our neighborhooding seemed to be well within the spatial tolerance of hail positioning within these resolved storms and thus produced large CSI values. The performance then decreased considerably in these neighborhooded plots for the final timesteps as the model had to rely more heavily on the WoFS forecasted evolution of convection, which inevitably became more erroneous with time. In scaling terms, this resulted in a large distinction in scale between the model performance/baseline difference across all pixelwise and neighborhooded max CSI plots that contained the ideal 3D U-Net. For timestep 00, this can be seen as a jump from approximately a 3x difference in performance to approximately a 4x difference. However, because of the aforementioned WoFS convection quality decay limitation, the difference in performance between U-Net results and baselines is unchanged across the pixelwise and neighborhooded max CSI plots at the final forecast timestep.

From our case study figures a further result was observed. There is an evident decrease in the magnitude of our severe hail predicted probabilities at the end of each forecast period. This is consistent across all shown case studies (Figures 4.9, 4.11, 4.13, and 4.15). This is a known downside to using a 3 dimensional U-Net's third spatial dimension as a temporal dimension. In particular, this behavior was noted in (Bansal et al. 2022). This decrease in the magnitude of the probabilities is due to the same issue associated with U-Net's poor quality at the spatial edges of their output

images. A U-Net's translating kernel is unable to resolve pixels past the image's edge. As such, the output pixels at the edge of the image are less accurate because they cannot have all nearby spatial structures properly resolved through the network. For our applications, one of these "spatial" dimensions is taken up by the time dimension. Therefore, the "spatial" third-dimensional edge is replaced by the temporal edges of our forecast period (the start and end timesteps of our forecast period). Consequently, this drop off in performance commonly observed at the edges of our data patches is also present at the final timestep of our forecast period (it has less of an effect in the first timestep because high-quality observations are present).

This also explains the aforementioned issues with patch spatial boundaries. Storms that pass over these boundaries are prone to this kernel-sourced error. This limitation is compounded with the fact that the U-Net is incapable of resolving storms that are advected into the patch from outside during any timestep past when observations are present. These limitations and their possible solutions warrant further discussion in a future study.

As shown in Figure 4.7, our area under the receiver operating characteristic (ROC) curve (AUC) values align consistently with the max CSI values we have observed in our predictor/label and architecture comparison experiments. We can see a slight increase in AUC performance by the 00 minute mark, which aligns with the peak max CSI in each of those former experiments. As discussed, this peak was hypothesized to be associated with WoFS spinup times. However, the reliability estimates shown in Figure 4.6 indicate some unusual behavior. We again see that a peak of values is observed at the 00 minute mark due to WoFS spinup. However, this performance peak is shown to be associated with a more unreliable forecast that is over-predicting severe hail. We believe this may be because the model was encouraged to over-predict when an abundance of observations were recently present, before proceeding to reduce

the magnitude of its probabilities towards the end of the forecast period. It should be noted that WoFS, like all other ensembles, is underdispersive (or overconfident) (Romine et al. 2014). This too requires additional study.

## 4.5 Case Studies

Case studies drawn from our test set are critical to confirm the validity of the results discussed throughout the sections outlining our test set evaluation. Additionally, they offer a view into what the output of these models may look like to a meteorologist in an operational setting. Our test set was produced through random sampling so there is no ordering by time across case studies. All 3 case studies were chosen semi-randomly from our test set (which was necessary as the test set was produced through our shuffling method, so there was only a small degree of artificial control possible in this selection process). However, despite this randomness, we chose dates that ensured both success and failure cases would be present by throwing out some of the dates initially chosen by random selection. These cases are pointed out in the following subsections. In the end, this selected process resulted in the 3 dates of May 18, 2017 (with WoFS initialization time at 19:00 UTC), May 19, 2018 (with WoFS initialization time at 19:00 UTC), and May 28, 2019 (with WoFS initialization time at 22:30 UTC).

All case studies are represented by 2 figures that contain several plots which display a section of each case study's WoFS domain. The first figure displays the 45 minute swath of our machine learning model's output after the 15 minute observation data period. Our model is represented by shades of green, while the WoFS deterministic >1 inch diameter HAILCAST swath is marked in blue, and the observed GridRad MESH >1 inch swath is red. These swath figures serve to give an overview of the model's performance relative to WoFS performance and observations for the most

useful 45 minutes of the forecast period. The second set of figures are all made up of 9 subplots that display 3 maps with 3 corresponding timesteps of the forecast period (ML forecast start + 00 mins, ML forecast start + 20 mins, ML forecast start + 40 mins). The bottom row's functionally is the same as the swath figures except it shows performance at discrete timesteps. The top row shows our forecasted probability of severe hail (black contours) relative to WoFS storm positioning (colored composite reflectivity). The middle row displays our forecasted probability of severe hail (black contours) relative to real GridRad composite reflectivity observed at each of the 3 given timesteps in the forecast period. All case study figures contain hail reports in the form of black circles for non-significant severe hail and black stars for significant severe hail. The hail reports in the swath figures are for all 45 minutes, while the hail reports in the timestep figures represent all reports from the previous 10 minutes to 10 minutes in the future.

### 4.5.1 Case Study 1 (May 18, 2017)

The first case study selected was a severe hail outbreak in western Oklahoma and far northwestern Texas on May 18, 2017, with WoFS initialization time at 19:00 UTC. During the forecast hour of this case study, 2 dominant storms near the southwestern corner of Oklahoma produced considerable quantities of severe hail and significant severe hail with several associated storm reports. This case study is outlined in Figures 4.8 and 4.9. The plots in Figure 4.9 reveal the ongoing convection resolved by WoFS at 00, 20, and 40 minutes past ML forecast start. All shown reflectivity in this particular plot is from the WoFS ensemble member that has been used as environmental variable input to our machine learning model. These plots also display the contours of our

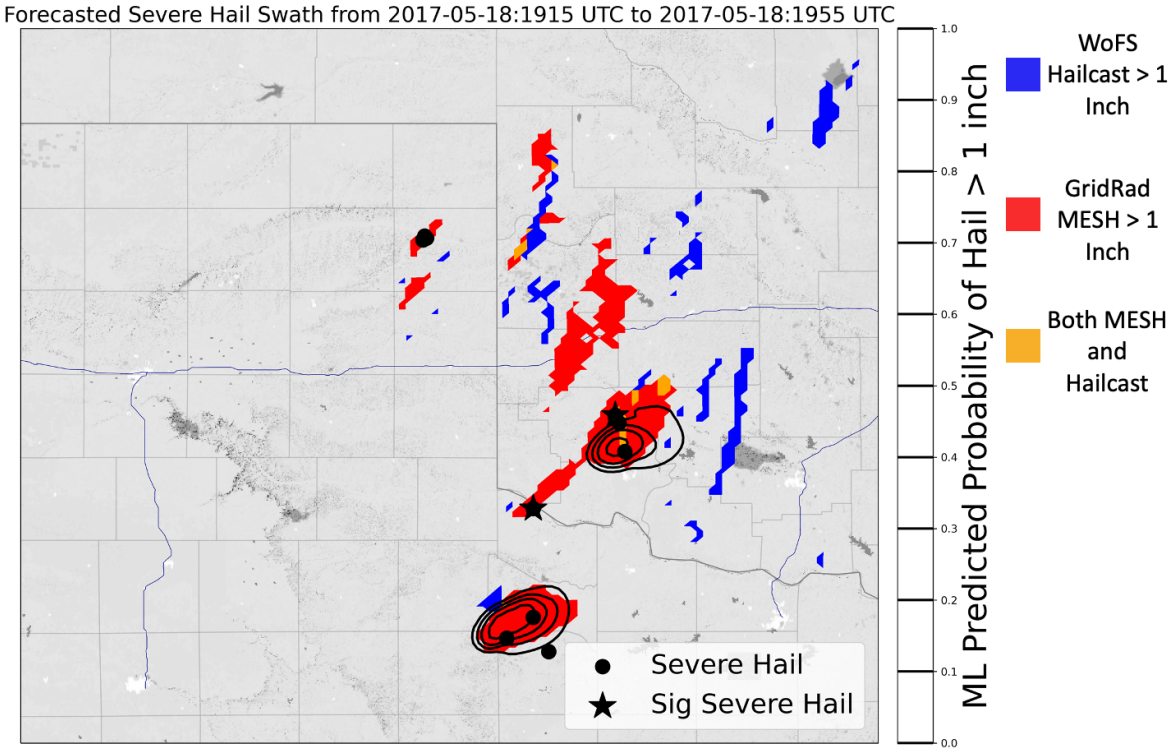



Figure 4.8: A map of western Oklahoma and northwestern Texas on May 18, 2017 starting at 19:15 UTC. A 45 minute swath of the forecasted probability of severe hail produced by our best model (ML output) is overlaid in black contours. HAILCAST >1 inch from a single WoFS ensemble member is displayed in blue. GridRad MESH >1 inch is displayed in red. Any significant or non-significant severe hail report that occurs during this 45 minute period is displayed with a black star or black circle respectively.

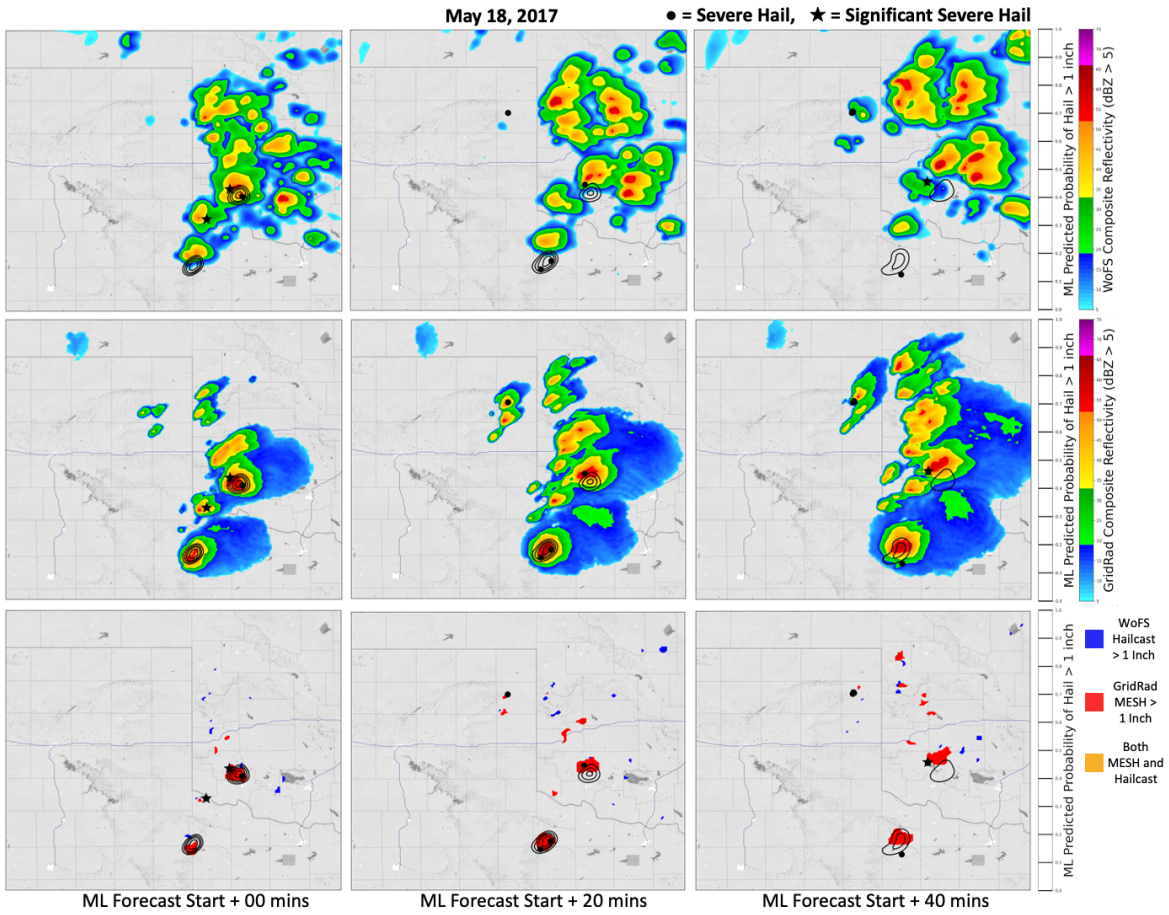



Figure 4.9: 3 maps of western Oklahoma and northwestern Texas on May 18, 2017 at 19:15 UTC, 19:30 UTC, and 19:55 UTC. This figure is made up of 3 distinct types of maps (the rows) with 3 shown timesteps (the columns) in the forecast period, all past the first valid timestep. The first map (top row) displays WoFS composite reflectivity from 1 ensemble member with our best model's predicted probability of severe hail overlaid as black contours ranging from 0 to 1.0. The second map (middle row) displays GridRad composite reflectivity with our best model's predicted probability of severe hail overlaid as black contours ranging from 0 to 1.0. The final map (bottom row) displays our best model's predicted probability of severe hail against the other two hail surrogates. All storm reports in each map are from the slice of time between 10 minutes before the timestep to 10 minutes after.

machine learning model's forecasted probability of hail greater than or equal to 1 inch in diameter (severe hail).

An immediately striking observation that can be found from these plots is the positioning of the machine learning's forecast relative to the WoFS storms. The position of real storms can be seen from the position of the storm reports that overlay the reflectivity and forecasted probabilities. Despite the WoFS forecast predicting a large set of storms, the machine learning forecast only predicts two severe hail producing storms. Additionally, the machine learning forecast predicts severe hail on a track that is spatially separate to the storm tracks predicted by WoFS. All WoFS storms have a strong northeastern component to their motion. In contrast, our forecasted severe hail contours possess a stronger eastward component of motion. Thus, our model appears to spatially align closer to the position of the various storm reports clustered near these two cells. This is especially evident for the southern of the two cells. The WoFS version of this cell clearly moves rapidly to the northeast away from the center of the severe hail reports. It also seems to predict the cell's rapid decay despite the real cell producing numerous severe hail reports at the forecast time that WoFS predicts this decay.

This event is further analyzed with the bottom row of Figure 4.9. This plot serves to highlight the differences between severe hail forecasted by the machine learning model (green contours) and the severe hail forecasted by WoFS (blue pixels). It also displays severe hail observed by GridRad MESH (red pixels). The disparity between the WoFS performance and the machine learning performance for the southern cell is especially highlighted in this plot. The WoFS hail barely predicts anything near this cell despite a strong GridRad MESH signal. Conversely, the machine learning forecast again performed well around this southern cell according to this plot.

This plot also reinforced the partial success of the machine learning model around the northern cell. The machine learning probabilities were spatially closer to, and of a more similar magnitude to the GridRad MESH signal for this cell. However, an offset was observed when comparing the exact position of the GridRad MESH pixels to the position of our forecasted pixels. Our forecast predicted severe hail further south than what was observed by GridRad MESH. Despite this, the latest hail reports agree more closely with what the machine learning model predicted. A possible explanation of this is that the radar-derived MESH does not correctly align with the updraft and consequently the hail of the storm in this example. It is important to recall that MESH is a surrogate for real hail (Witt et al. 1998). Another explanation is that the machine learning forecast for this storm diverges from reality, while coincidentally the more southern hail report arises from human error or from the discussed rural bias issues.

The final result that can be derived from this case study is the failure case observed in the northern half of the domain. Several tightly clustered hail reports were observed in the northeastern corner of the Texas panhandle. Since there were several hail reports in this cluster it appears prudent to assume a storm was legitimately present. Despite this, neither WoFS or the machine learning model correctly predicted this storm at any point in the forecast period. A possible explanation for this is that the storm did not have any ongoing convection in the first 15 minutes after WoFS initialization. Therefore, the machine learning did not have any ongoing storm activity of any kind to advect forward in its forecast. This theory is supported by the middle row of Figure 4.9, which reveals that real convection was not occurring at this location in the early stages of the forecast period. Without this proper convective initialization, both WoFS and our machine learning model failed to accurately forecast the hail in this area. This highlights a critical weakness of a nowcasting model that relies upon early existing convection. Ideally, a future nowcasting model will be capable of using both existing

storm observations and simulated future convective initialization to predict severe hail in the short term.

### 4.5.2 Case Studies 2 and 3 (May 19, 2018 and May 28, 2019)

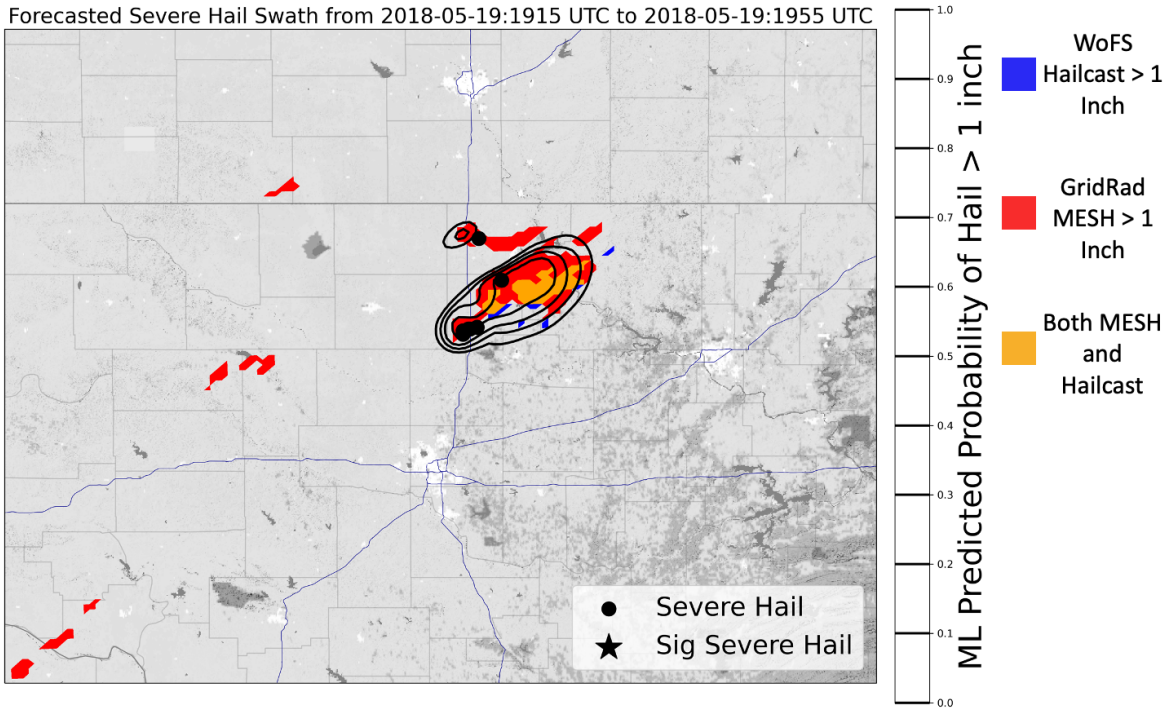



Figure 4.10: A map of northern Oklahoma on May 19, 2018 starting at 19:15 UTC. A 45 minute swath of the forecasted probability of severe hail produced by our best model (ML output) is overlaid in black contours. HAILCAST >1 inch from a single WoFS ensemble member is displayed in blue. GridRad MESH >1 inch is displayed in red. Any significant or non-significant severe hail report that occurs during this 45 minute period is displayed with a black star or black circle respectively.

The second case study was on the severe hail event in northern Oklahoma that occurred on May 19, 2018. This case study again showed the success of the machine learning model in the vicinity of all hail reports made during the forecast period.

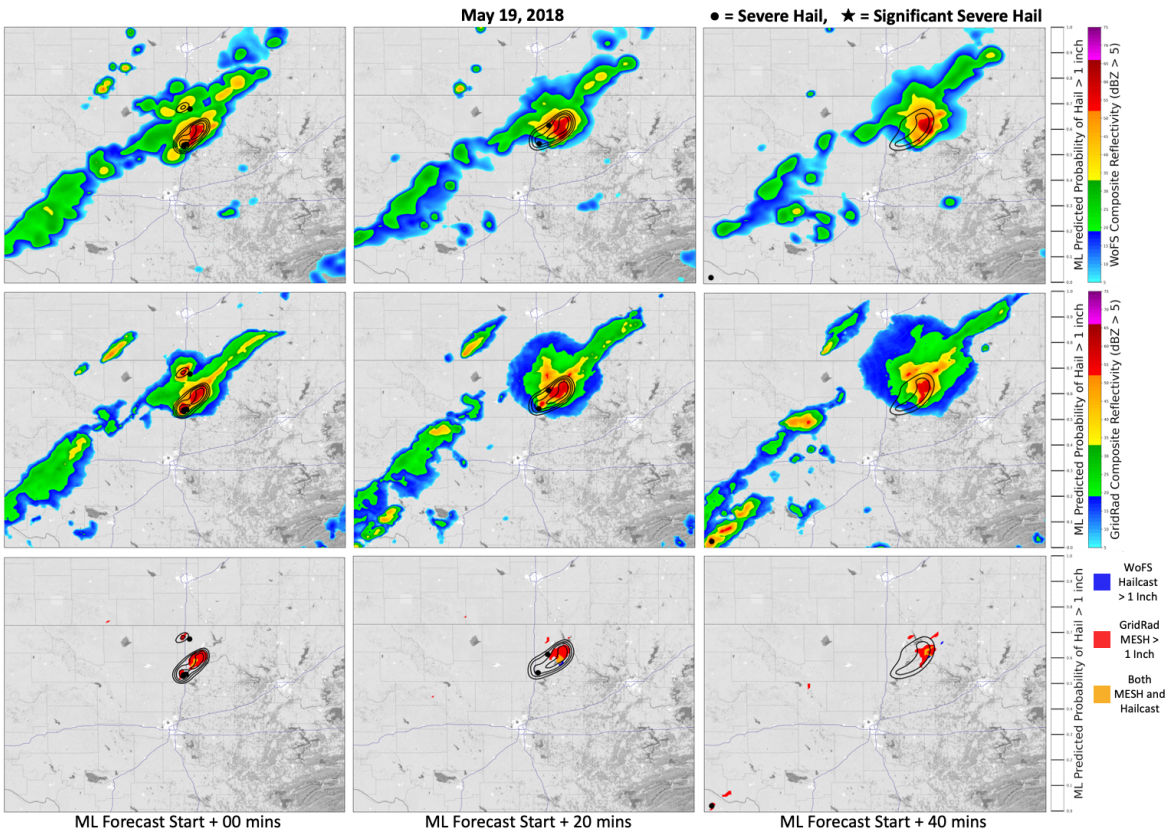



Figure 4.11: 3 maps of northern Oklahoma on May 19, 2018 at 19:15 UTC, 19:30 UTC, and 19:55 UTC. This figure is made up of 3 distinct types of maps (the rows) with 3 shown timesteps (the columns) in the forecast period, all past the first valid timestep. The first map (top row) displays WoFS composite reflectivity from 1 ensemble member with our best model's predicted probability of severe hail overlaid as black contours ranging from 0 to 1.0. The second map (middle row) displays GridRad composite reflectivity with our best model's predicted probability of severe hail overlaid as black contours ranging from 0 to 1.0. The final map (bottom row) displays our best model's predicted probability of severe hail against the other two hail surrogates. All storm reports in each map are from the slice of time between 10 minutes before the timestep to 10 minutes after.

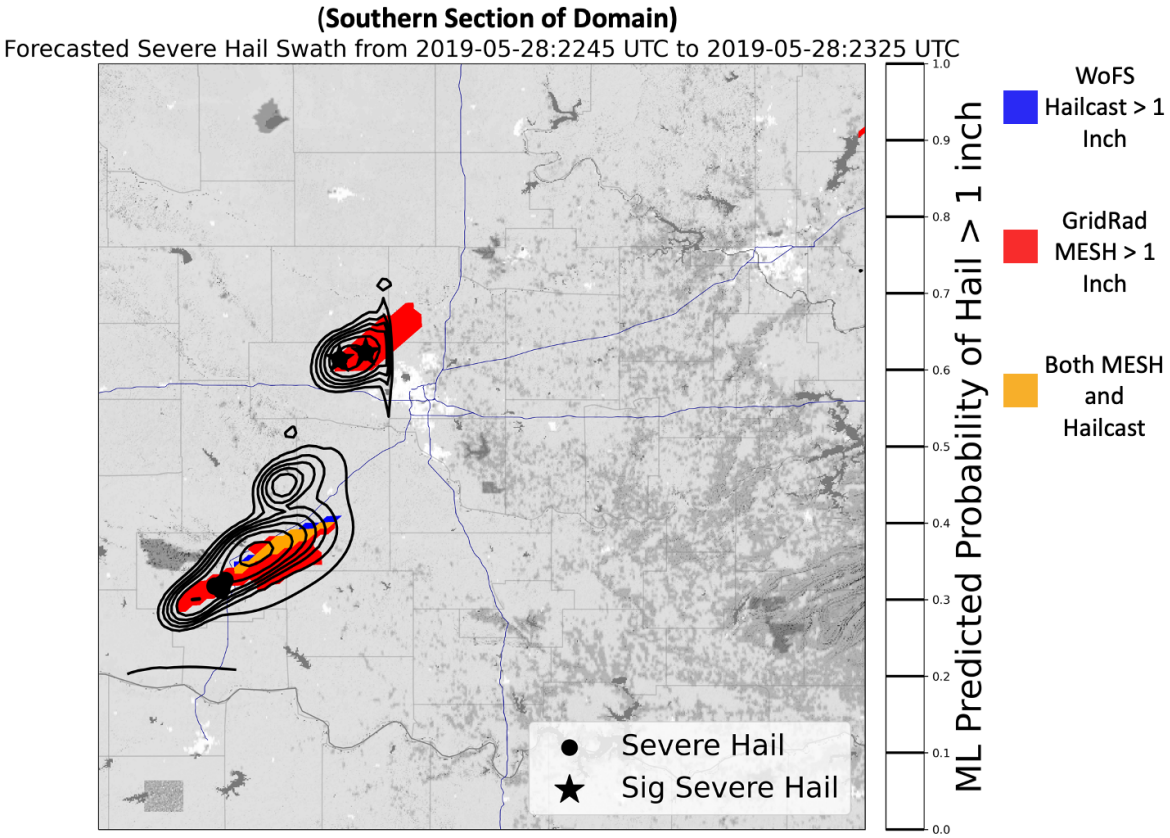



Figure 4.12: A map of central Oklahoma on May 28, 2019 starting at 22:45 UTC. A 45 minute swath of the forecasted probability of severe hail produced by our best model (ML output) is overlaid in black contours. HAILCAST >1 inch from a single WoFS ensemble member is displayed in blue. GridRad MESH >1 inch is displayed in red. Any significant or non-significant severe hail report that occurs during this 45 minute period is displayed with a black star or black circle respectively.

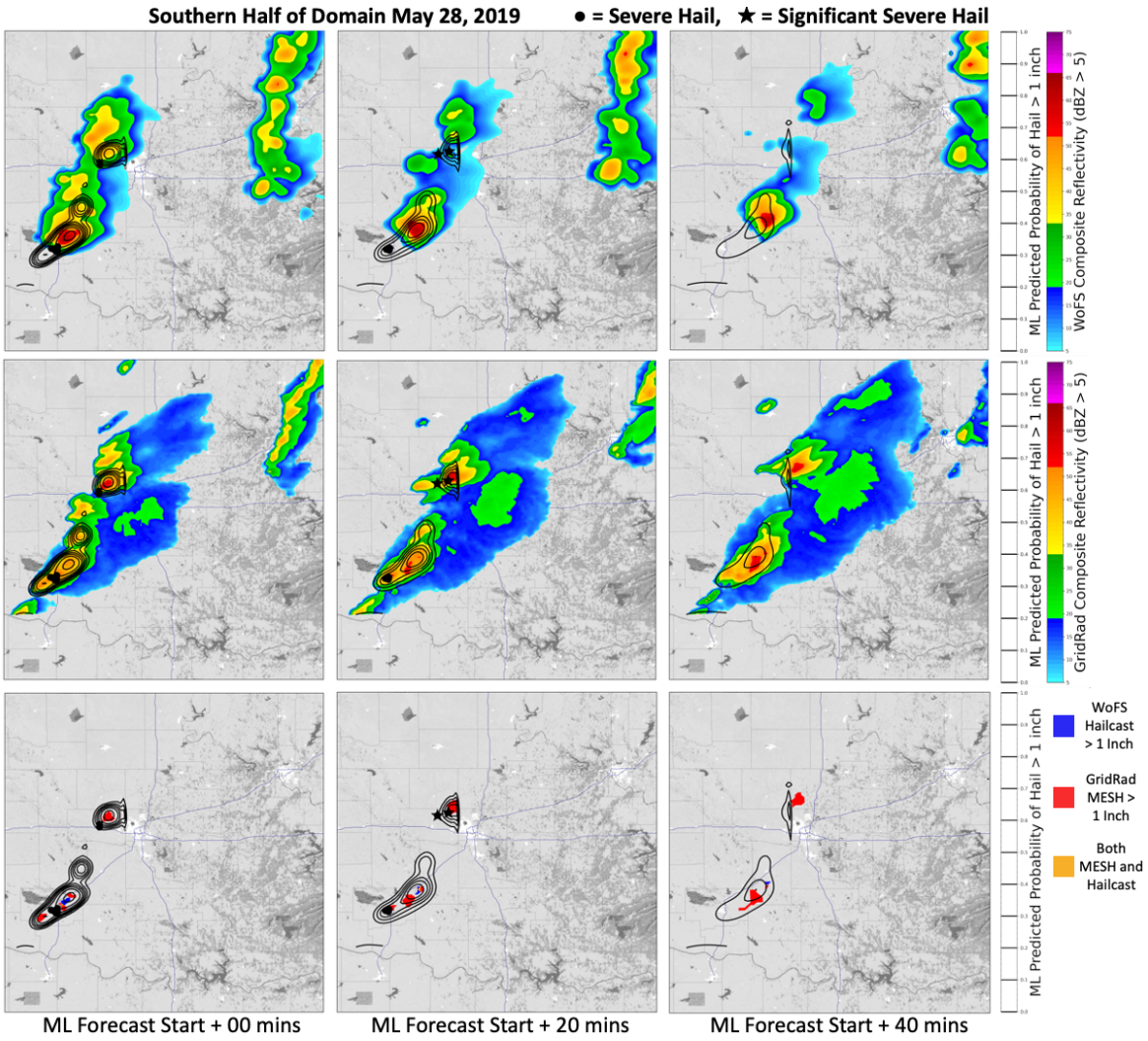



Figure 4.13: 3 maps of central Oklahoma on May 28, 2019 at 22:45 UTC, 23:00 UTC, and 23:15 UTC. This figure is made up in the same format as figure 4.9.

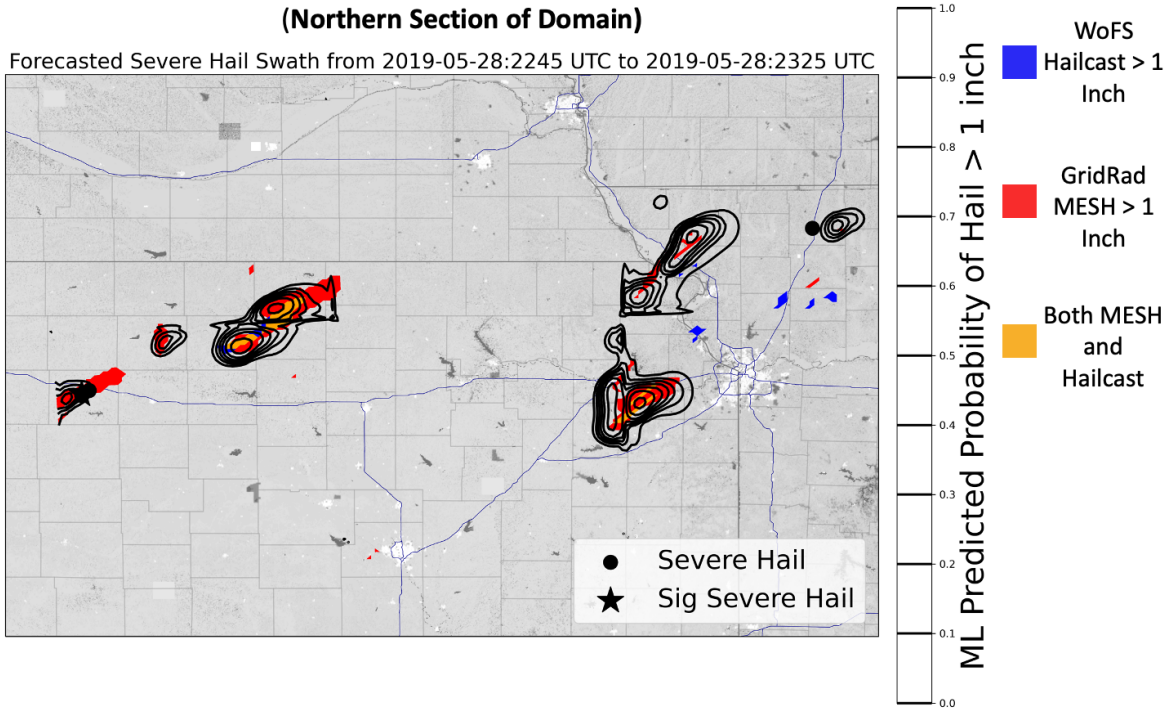



Figure 4.14: A map of northern Kansas on May 28, 2019 starting at 22:45 UTC. A 45 minute swath of the forecasted probability of severe hail produced by our best model (ML output) is overlaid in black contours. HAILCAST >1 inch from a single WoFS ensemble member is displayed in blue. GridRad MESH >1 inch is displayed in red. Any significant or non-significant severe hail report that occurs during this 45 minute period is displayed with a black star or black circle respectively.

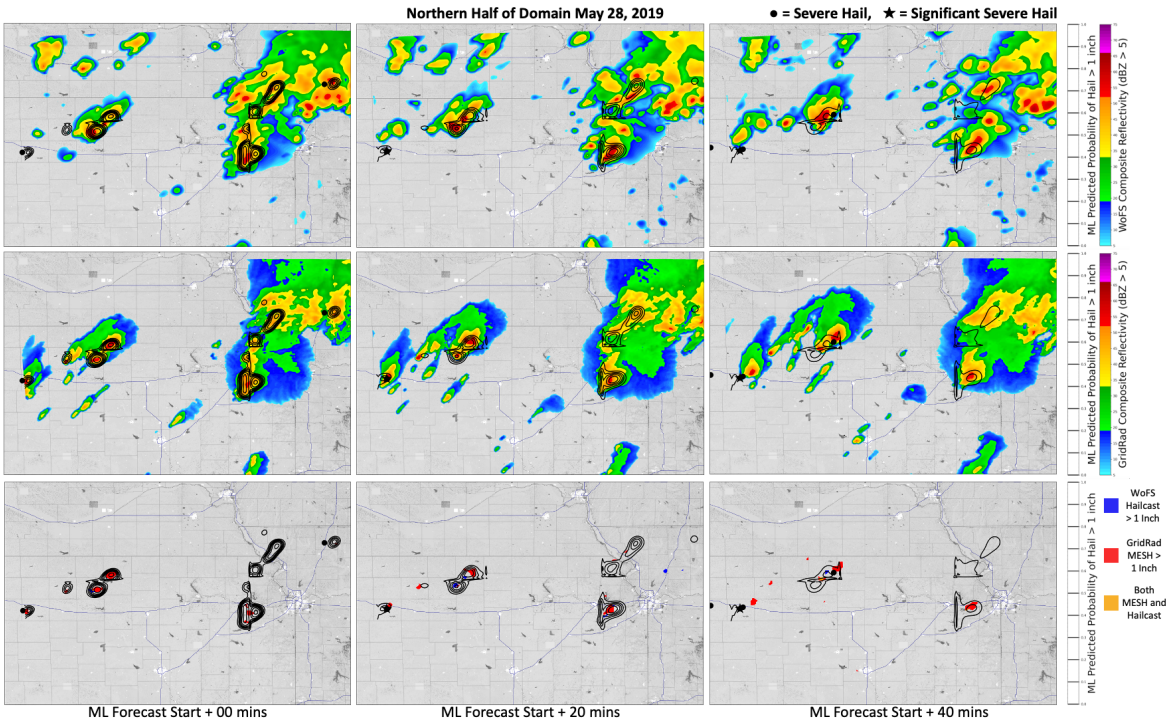



Figure 4.15: 3 maps of northern Kansas on May 28, 2019 at 22:45 UTC, 23:00 UTC, and 23:15 UTC. This figure is made up in the same format as figure 4.9.

This case study was outlined in Figures 4.10 and 4.11. A clearly dominant pair of cells was located in northern Oklahoma with corresponding hail reports and GridRad MESH. As in case study 1, the machine learning model forecasted severe hail over the location of these hail reports. However, the magnitude of the predicted probabilities in the northern storm shrank to a small degree, but this aligns with the death of the real storm and thus this behavior can be considered positive. A further result can be observed from the hail comparison plots, which show that the WoFS hail forecasted the southern storm fairly well, while missing the northern storm. However, once again, the machine learning model correctly forecasted this northern storm that WoFS missed.

The final result pulled from this case study can be seen in the western half of the domain. In this area, several isolated groupings of GridRad MESH can be observed. These areas of radar returns did not have any corresponding hail reports nor did WoFS initialize any strong storms in their vicinity. It appears that the convective activity in this area was not exceedingly severe and that these storms did not physically warrant a strong severe hail forecast. Therefore, this section of the case study can be considered an example of a true negative forecast from our model.

Our final case study occurred on May 28, 2019 in central Oklahoma and northern Kansas. This event had two distinct areas of significant convection in these two states and therefore it warranted two sets of figures. The first set of figures (4.12 and 4.13) shows the Oklahoma storms. These figures are once again primarily defined by a set of successful forecast hits. Figure 4.12 reveals that our forecasted severe hail aligns well with location of both the GridRad MESH labels and hail reports for the two primary cells over central Oklahoma. There is a small area of lower severe hail probabilities immediately to the north of the southern cell that does not produce significant hail. This small area of forecasted hail could be considered a minor false positive example,

however, this smaller sibling cell appears to be the remnants of a left-mover and the magnitude of the forecasted probabilities decay rapidly.

When examining the top row of 4.13 the skillful performance of our machine learning model relative to WoFS is highlighted. Both of the significant cells in central Oklahoma near the Oklahoma city metropolitan area (outlined in white) have severe hail that is well forecasted by our models. This is evident from the positioning of hail reports and GridRad MESH relative to our machine learning model's forecasted areas of severe hail. It also appears from this figure that the positioning of our model's forecast once again outperforms WoFS positioning. WoFS reflectivity for the northern storm passes too far north of the actual severe hail, however, our model does not. Additionally, the southern storm's WoFS reflectivity is slightly fast relative to storm reports, however, the contours of our model successfully account for this and partially remain behind. In addition to these successes, there are some limited areas of WoFS reflectivity to the East that are also completely ignored as desired.

The other set of figures representing the May 28, 2019 event visualize the Kansas area of the domain (Figures 4.14 and 4.15). The storms in this area represented the greatest challenge for our machine learning model across all discussed case studies. These plots highlight several true positive storms but also several false positive storms. The storm in the far northeastern corner of the plots is technically a true positive example, however, the hail report occurs at almost exactly the 00 minute timestep which is not a useful forecast. A second true positive example occurs on the western edge of the plots. In this example, the machine learning forecasts the hail that produces the storm reports well in advance of the hail occurring.

In this case study, two possible false positive examples occurred to the west and northwest of Kansas City (large white outlined city in the eastern half of plots). The northern of the two storms was a clear false positive as there were no hail reports

nor a lot of GridRad MESH returns in this area. The southern of the two had more unusual behaviour. This storm did not produce a hail report at any point during the forecast period. However, there was a considerable swath of GridRad MESH in this vicinity. This may imply that this is an example of missing or erroneous storm reports. Alternatively, this may be an area of relatively poor MESH quality as both these data sources are error prone surrogates for hail. A final factor that further complicates the prediction of storms in this area is the clearly evident U-Net patch boundary running through the center of these forecasted storms. This can be seen as the unusual vertically stretched lines in the prediction contours around these storms (middle and top rows of Figure 4.15). The U-Net is unable to advect storms through the boundaries of its patches and as such performance tends to be poorer near these edges. This may also be an explanation for the generally erroneous behavior in this entire vicinity.

# Chapter 5

# Discussion and Conclusions

The two main contributions of this thesis are 1) developing and testing a hail nowcasting machine-learning algorithm that could be used in real-time and 2) demonstrating the importance of incorporating real-time observations in nowcasting problems. It was clear from the ablation experiment on the Vaisala lighting predictor that real-time observations were critical to the performance of our machine learning model. As discussed in our results chapter, this predictor was so impactful to our model, that the predictor entirely altered the model's primary behavior. Throughout the hour-long forecast period the introduction of this lightning predictor consistently increased model performance by several factors.

This observation-importance behavior has also been observed in several other studies (Czernecki et al. 2019, Leinonen et al. 2022). One such example is the model produced for (Czernecki et al. 2019), where CSI performance values greatly increased when radar observations and ERA5 reanalysis were combined for use in a large hail machine learning model. When combined, these sources nearly doubled the CSI of models that were trained on only one of the two (Czernecki et al. 2019). Changes in performance of this scale are significant and the fact that several independent studies have now produced this result motivates future research on this topic.

Some experiments performed for this thesis did not contain any observation data and were only trained on WoFS predictors. As such the machine learning models trained on this data had their performance constrained by the performance of WoFS.

This resulted in our model performance having a strong correlation with "dataset hail climatology", as WoFS performance could vary from year to year based upon real climatology or WoFS architecture changes. Thus, when we split our data into training, validation, and testing sets, performance would vary strongly when the splits were moved. This highlighted the need for our storm-day clustering system and subsequent sampling methods used in our dataset generation. Any future machine learning studies that rely heavily on NWP data for predictors should be careful to study the intricacies of all used datasets and avoid the use of date delineated data partitions.

In conclusion, all objectives set out for this thesis have been successfully met. Our primary objective was to create a quality severe hail nowcasting model using a machine learning framework. We considered a quality hail nowcasting model to consist of any model that offered a substantial increase in performance over well-established hail baselines for all timesteps in our forecast period. These baselines consisted of WoFS HAILCAST ensemble probability of severe hail and updraft helicity-based logistic regression. We created 3 architecture experiments to compare the effectiveness of different U-Net architectures for this severe hail problem. Architecture experiment 1 was used as a baseline and only had a single basic U-Net with no added temporal logic. Architecture experiment 2 used multiple 2D U-Nets to predict all forecast timesteps. Finally, architecture experiment 3 used a single 3D U-Net to predict all timesteps at once with an added time dimension. Nearly all forecasts produced by our machine learning models used in architecture experiments 2 and 3 outperformed the two hail baselines for nearly all timesteps in the forecast period (see Figure 4.5). The one exception to this being the performance of the deterministic U-Net created by the ablation of the lightning observations in the final two timesteps of the forecast period.

The secondary objective of this thesis was to investigate the effect on model performance when adding a real-time observation predictor to a machine learning model

already trained with NWP predictors. This objective was met with the lightning ablation results. That ablation experiment clearly highlights the strong contributions that real-time observations can bring to the nowcasting of severe hail. All our models that used the Vaisala lightning data vastly outperformed those that did not. Additionally, the introduction of these observations to our machine learning models transformed the overall behavior of the models. We theorized that our models that used the lightning observations switched the predictor used to drive their storm initialization from WoFS composite reflectivity to the lightning data itself. With this change, the WoFS predictors were only used as environmental variables necessary for the forecasting of storm evolution. These were the originally stated goals of both these datasets. Evidence supporting this theory can be seen in Figure 4.4, where the ensemble distribution only benefits the models that do not contain lightning observations. This is because the ensemble spread is only useful for dealing with the WoFS convection quality issue, it does not affect the quality of the WoFS environmental variables. Therefore, only our machine learning models that used WoFS data (and not lightning data) for their storm initialization predictor show an improvement in ensemble performance in all max CSI plots.

Our third and final objective was to test the forecasting performance of three distinct architecture experiments that were each made up of various U-Nets. These 3 architecture experiments were outlined in section 3.2 and with Figures 3.1 and 3.2. The comparison process was outlined in Figure 4.5. This also proved to be a success, as a clearly dominant architecture was discovered. This information may be be useful to future researchers considering machine learning model options. These results also align well with the findings of similar recent studies on the use of U-Nets for weather forecasting applications (Bansal et al. 2022).

Although this thesis has produced a hail nowcasting model of sufficient quality, future studies are required to further evaluate the model's performance. Additionally, our particular real-time observations dataset was selected for its extensive global domain. With the success of its use in this hybrid machine learning and NWP environmental variable framework, the envisioned global upscaling of this model seems to be a logical next step for a future study. A global NWP model such as the IBM GRAF model could easily supply environment variables at the resolution of WoFS for a global scale. This, in combination with the global Vaisala lightning data network, could be used in a machine learning framework similar to ours with correspondingly sufficient performance.

Additionally, future studies are required to test further machine learning architectures for use in this problem. In particular, architectures optimized to resolve time could offer further performance improvements for this problem. One such architecture is found in using the same U-Net-only approach but with a more refined loss function. It is theorized that one could create a loss function that weights predictors differently depending on the timestep. This is advantageous to an observations/NWP hybrid predictor approach to a nowcasting problem such as ours because it enables a researcher to enforce rules upon the value of a predictor as a function of time. Ideally, this would help control the behavior we observed where a 3D U-Net would either over favor the observation predictors or the NWP predictors throughout the whole forecast period as opposed to varying this favoritism with time. Another such optimized time-resolving model is a long short-term memory (LSTM) network (Gers et al. 2000). These models are specifically engineered for this task. Additional study is warranted on the use of this particular network.

# Reference List


Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, 2015: TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. URL https://www.tensorflow.org/

Adams-Selin, R. D., A. J. Clark, C. J. Melick, S. R. Dembek, I. L. Jirak, and C. L. Ziegler, 2019: Evolution of wrf-hailcast during the 2014–16 noaa/hazardous weather testbed spring forecasting experiments. *Weather and Forecasting*, **34**, 61 – 79. URL https://journals.ametsoc.org/view/journals/wefo/34/1/waf-d-18-0024_1.xml

Adams-Selin, R. D. and C. L. Ziegler, 2016: Forecasting hail using a one-dimensional hail growth model within wrf. *Monthly Weather Review*, **144**, 4919 – 4939. URL https://journals.ametsoc.org/view/journals/mwre/144/12/mwr-d-16-0027.1.xml

Albawi, S., T. A. Mohammed, and S. Al-Zawi, 2017: Understanding of a convolutional neural network. *2017 International Conference on Engineering and Technology (ICET)*, 1–6.

Allen, D. M., 1974: The relationship between variable selection and data agumentation and a method for prediction. *Technometrics*, **16**, 125–127. URL https://www.tandfonline.com/doi/abs/10.1080/00401706.1974.10489157

Allen, J. T., I. M. Giammanco, M. R. Kumjian, H. Jurgen Punge, Q. Zhang, P. Groenemeijer, M. Kunz, and K. Ortega, 2020: Understanding hail in the earth system. *Reviews of Geophysics*, **58**, e2019RG000665, e2019RG000665 10.1029/2019RG000665. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019RG000665

Allen, J. T., D. J. Karoly, and G. A. Mills, 2011: A severe thunderstorm climatology for australia and associated thunderstorm environments. *Australian Meteorological and Oceanographic Journal*, **61**, 143.

Allen, J. T. and M. K. Tippett, 2015: The characteristics of united states hail reports: 1955-2014. *E-Journal of Severe Storms Meteorology*, **10**, 1–31.

Bansal, A. S., Y. Lee, K. Hilburn, and I. Ebert-Uphoff, 2022: Tools for extracting spatio-temporal patterns in meteorological image sequences: From feature engineering to attention-based neural networks.

Beauchemin, M., 2023: How insurance companies save millions with proactive weather alerts.
URL https://www.tomorrow.io/blog

Billet, J., M. DeLisi, B. G. Smith, and C. Gates, 1997: Use of regression techniques to predict hail size and the probability of large hail. *Weather and Forecasting*, **12**, 154 – 164.
URL https://journals.ametsoc.org/view/journals/wefo/12/1/1520-0434_1997_012_0154_uorttp_2_0_co_2.xml

Birant, D. and A. Kut, 2007: St-dbscan: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, **60**, 208–221, intelligent Data Mining.
URL https://www.sciencedirect.com/science/article/pii/S0169023X06000218

Blair, S. F., J. M. Laflin, D. E. Cavanaugh, K. J. Sanders, S. R. Currens, J. I. Pullin, D. T. Cooper, D. R. Deroche, J. W. Leighton, R. V. Fritchie, M. J. M. II, B. T. Goudeau, S. J. Kreller, J. J. Bosco, C. M. Kelly, and H. M. Mallinson, 2017: High-resolution hail observations: Implications for nws warning operations. *Weather and Forecasting*, **32**, 1101 – 1119.
URL https://journals.ametsoc.org/view/journals/wefo/32/3/waf-d-16-0203_1.xml

Brimelow, J. C., G. W. Reuter, and E. R. Poolman, 2002: Modeling maximum hail size in alberta thunderstorms. *Weather and Forecasting*, **17**, 1048 – 1062.
URL https://journals.ametsoc.org/view/journals/wefo/17/5/1520-0434_2002_017_1048_mmhsia_2_0_co_2.xml

Calhoun, K. M., T. M. Smith, D. M. Kingfield, J. Gao, and D. J. Stensrud, 2014: Forecaster use and evaluation of real-time 3dvar analyses during severe thunderstorm and tornado warning operations in the hazardous weather testbed. *Wea. Forecasting*, **29**, 601–613.

Cecil, D. J., 2009: Passive microwave brightness temperatures as proxies for hailstorms. *Journal of Applied Meteorology and Climatology*, **48**, 1281 – 1286.
URL https://journals.ametsoc.org/view/journals/apme/48/6/2009jamc2125.1.xml

Changnon, S. A., D. Changnon, E. R. Fosse, D. C. Hoganson, R. J. Roth, and J. M. Totsch, 1997: Effects of recent weather extremes on the insurance industry: Major implications for the atmospheric sciences. *Bulletin of the American Meteorological Society*, **78**, 425 – 436.
URL https://journals.ametsoc.org/view/journals/bams/78/3/1520-0477_1997_078_0425_eorweo_2_0_co_2.xml

Chase, R. J., D. R. Harrison, A. Burke, G. M. Lackmann, and A. McGovern, 2022: A machine learning tutorial for operational meteorology. part i: Traditional machine learning. *Weather and Forecasting*, **37**, 1509 – 1529.

URL https://journals.ametsoc.org/view/journals/wefo/37/8/WAF-D-22-0070.1.xml

Chase, R. J., D. R. Harrison, G. M. Lackmann, and A. McGovern, 2023: A machine learning tutorial for operational meteorology, part II: Neural networks and deep learning. *Weather and Forecasting*.
URL https://doi.org/10.1175\%2Fwaf-d-22-0187.1

Cintineo, J. L., T. M. Smith, V. Lakshmanan, H. E. Brooks, and K. L. Ortega, 2012: An objective high-resolution hail climatology of the contiguous united states. *Weather and Forecasting*, **27**, 1235 – 1248.
URL https://journals.ametsoc.org/view/journals/wefo/27/5/waf-d-11-00151_1.xml

Cummins, K. L., J. A. Cramer, C. J. Biagi, E. P. Krider, J. Jerauld, M. A. Uman, and V. A. Rakov, 2006: 6.1. the us national lightning detection network: post-upgrade status. *Second Conference on Meteorological Applications of Lightning Data*.

Czernecki, B., M. Taszarek, M. Marosz, M. Półrolniczak, L. Kolendowicz, A. Wyszogrodzki, and J. Szturc, 2019: Application of machine learning to large hail prediction - the importance of radar reflectivity, lightning occurrence and convective parameters derived from era5. *Atmospheric Research*, **227**, 249–262.
URL https://www.sciencedirect.com/science/article/pii/S0169809519300900

Dennis, E. J. and M. R. Kumjian, 2017: The impact of vertical wind shear on hail growth in simulated supercells. *Journal of the Atmospheric Sciences*, **74**, 641 – 663.
URL https://journals.ametsoc.org/view/journals/atsc/74/3/jas-d-16-0066.1.xml

Earnest, B., A. McGovern, I. L. Jirak, and C. Karstens, 2023: Examining the role of the wildfire triangle in predicting wildfire occurrence for conus with the unet3+ model, aMS 2023.
URL https://ams.confex.com/ams/103ANNUAL/meetingapp.cgi/Paper/419373

Flora, M. L., C. K. Potvin, P. S. Skinner, S. Handler, and A. McGovern, 2021: Using machine learning to generate storm-scale probabilistic guidance of severe weather hazards in the warn-on-forecast system. *Monthly Weather Review*, **149**, 1535 – 1557.
URL https://journals.ametsoc.org/view/journals/mwre/149/5/MWR-D-20-0194.1.xml

Foster, D. S. and F. C. Bates, 1956: A hail size forecasting technique. *Bulletin of the American Meteorological Society*, **37**, 135 – 141.
URL https://journals.ametsoc.org/view/journals/bams/37/4/1520-0477-37_4_135.xml

Gagne, D. J., S. E. Haupt, D. W. Nychka, and G. Thompson, 2019: Interpretable deep learning for spatial analysis of severe hailstorms. *Monthly Weather Review*, **147**, 2827 – 2845.

URL https://journals.ametsoc.org/view/journals/mwre/147/8/mwr-d-18-0316.1.xml

Gagne, D. J., A. McGovern, S. E. Haupt, R. A. Sobash, J. K. Williams, and M. Xue, 2017: Storm-based probabilistic hail forecasting with machine learning applied to convection-allowing ensembles. *Weather and Forecasting*, **32**, 1819 – 1840.
URL https://journals.ametsoc.org/view/journals/wefo/32/5/waf-d-17-0010_1.xml

Gallo, B. T., 2017: Breaking new ground in severe weather prediction: The 2015 noaa/hazardous weather testbed spring forecasting experiment. *Wea. Forecasting*, **32**, 1541–1568.

Gao, J., 2013: A real-time weather-adaptive 3dvar analysis system for severe weather detections and warnings with automatic storm positioning capability. *Wea. Forecasting*, **28**, 727–745.

Gensini, V. A., C. Converse, W. S. Ashley, and M. Taszarek, 2021: Machine learning classification of significant tornadoes and hail in the united states using era5 proximity soundings. *Weather and Forecasting*, **36**, 2143 – 2160.
URL https://journals.ametsoc.org/view/journals/wefo/36/6/WAF-D-21-0056.1.xml

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

Gunturi, P. and M. Tippett, 2017: Managing severe thunderstorm risk: Impact of enso on us tornado and hail frequencies. *Willis Re Inc*.

Hong, S.-Y. and J. Dudhia, 2012: Next-generation numerical weather prediction: Bridging parameterization, explicit clouds, and large eddies. *Bulletin of the American Meteorological Society*, **93**, ES6–ES9.
URL http://www.jstor.org/stable/26218628

Hu, M. and M. Xue, 2007: Impact of configurations of rapid intermittent assimilation of wsr-88d radar data for the 8 may 2003 oklahoma city tornadic thunderstorm case. *Monthly Weather Review*, **135**, 507 – 525.
URL https://journals.ametsoc.org/view/journals/mwre/135/2/mwr3313.1.xml

Huang, H., L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, and J. Wu, 2020: Unet 3+: A full-scale connected unet for medical image segmentation.

Jewell, R. and J. Brimelow, 2009: Evaluation of alberta hail growth model using severe hail proximity soundings from the united states. *Weather and Forecasting*, **24**, 1592 – 1609.
URL https://journals.ametsoc.org/view/journals/wefo/24/6/2009waf2222230_1.xml

Johnson, A. W. and K. E. Sugden, 2014: Evaluation of sounding-derived thermodynamic and wind-related parameters associated with large hail events. *E-Journal of Severe Storms Meteorology*, **9**, 1–42.

Justin, A. D., C. Willingham, A. McGovern, and J. T. Allen, 2023: Toward operational real-time identification of frontal boundaries using machine learning. *Artificial Intelligence for the Earth Systems*, **2**, e220052.
URL https://journals.ametsoc.org/view/journals/aies/2/3/AIES-D-22-0052.1.xml

Kingma, D. P. and J. Ba, 2017: Adam: A method for stochastic optimization.

Kochkov, D., J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, and S. Hoyer, 2021: Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, **118**, e2101784118.
URL https://www.pnas.org/doi/abs/10.1073/pnas.2101784118

Labriola, J., N. Snook, Y. Jung, and M. Xue, 2019: Explicit ensemble prediction of hail in 19 may 2013 oklahoma city thunderstorms and analysis of hail growth processes with several multimoment microphysics schemes. *Monthly Weather Review*, **147**, 1193 – 1213.
URL https://journals.ametsoc.org/view/journals/mwre/147/4/mwr-d-18-0266.1.xml

Lagerquist, R., A. McGovern, C. R. Homeyer, D. J. Gagne, and T. Smith, 2020: Deep learning on three-dimensional multiscale data for next-hour tornado prediction. *Monthly Weather Review*, **148**, 2837 – 2861.
URL https://journals.ametsoc.org/view/journals/mwre/148/7/mwrD190372.xml

Lagerquist, R., A. McGovern, and T. Smith, 2017: Machine learning for real-time prediction of damaging straight-line convective wind. *Weather and Forecasting*, **32**, 2175 – 2193.
URL https://journals.ametsoc.org/view/journals/wefo/32/6/waf-d-17-0038_1.xml

Leinonen, J., U. Hamann, U. Germann, and J. R. Mecikalski, 2022: Nowcasting thunderstorm hazards using machine learning: the impact of data sources on performance. *Natural Hazards and Earth System Sciences*, **22**, 577–597.
URL https://nhess.copernicus.org/articles/22/577/2022/

Manzato, A., 2013: Hail in northeast italy: A neural network ensemble forecast using sounding-derived indices. *Weather and Forecasting*, **28**, 3 – 28.
URL https://journals.ametsoc.org/view/journals/wefo/28/1/waf-d-12-00034_1.xml

McGovern, A., R. J. Chase, M. Flora, D. J. Gagne, R. Lagerquist, C. K. Potvin, N. Snook, and E. Loken, 2023: A review of machine learning for convective weather. *Artificial Intelligence for the Earth Systems*, 1 – 61.

URL https://journals.ametsoc.org/view/journals/aies/aop/AIES-D-22-0077.1/AIES-D-22-0077.1.xml

McGovern, A., K. L. Elmore, D. J. Gagne, S. E. Haupt, C. D. Karstens, R. Lagerquist, T. Smith, and J. K. Williams, 2017: Using artificial intelligence to improve real-time decision-making for high-impact weather. *Bulletin of the American Meteorological Society*, **98**, 2073 – 2090.
URL https://journals.ametsoc.org/view/journals/bams/98/10/bams-d-16-0123.1.xml

Milbrandt, J. A. and M. K. Yau, 2005: A multimoment bulk microphysics parameterization. part i: Analysis of the role of the spectral shape parameter. *Journal of the Atmospheric Sciences*, **62**, 3051 – 3064.
URL https://journals.ametsoc.org/view/journals/atsc/62/9/jas3534.1.xml

Mohr, S. and M. Kunz, 2013: Recent trends and variabilities of convective parameters relevant for hail events in germany and europe. *Atmospheric Research*, **123**, 211–228, 6th European Conference on Severe Storms 2011. Palma de Mallorca, Spain.
URL https://www.sciencedirect.com/science/article/pii/S016980951200155X

Morrison, H., M. van Lier-Walqui, A. M. Fridlind, W. W. Grabowski, J. Y. Harrington, C. Hoose, A. Korolev, M. R. Kumjian, J. A. Milbrandt, H. Pawlowska, D. J. Posselt, O. P. Prat, K. J. Reimel, S.-I. Shima, B. van Diedenhoven, and L. Xue, 2020: Confronting the challenge of modeling cloud and precipitation microphysics. *Journal of Advances in Modeling Earth Systems*, **12**, e2019MS001689, e2019MS001689 2019MS001689.
URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019MS001689

Murillo, E. M. and C. R. Homeyer, 2019: Severe hail fall and hailstorm detection using remote sensing observations. *Journal of Applied Meteorology and Climatology*, **58**, 947 – 970.
URL https://journals.ametsoc.org/view/journals/apme/58/5/jamc-d-18-0247.1.xml

Murillo, E. M., C. R. Homeyer, and J. T. Allen, 2021: A 23-year severe hail climatology using gridrad mesh observations. *Monthly Weather Review*, **149**, 945 – 958.
URL https://journals.ametsoc.org/view/journals/mwre/149/4/MWR-D-20-0178.1.xml

Nelson, S. P., 1983: The influence of storm flow structure on hail growth. *J. Atmos. Sci.*, **40**, 1965–1983.

NSSL, 2021: .
URL https://www.nssl.noaa.gov/projects/wof/casestudies/hail-oktx-apr2021/

Ortega, K. L., 2018: Evaluating multi-radar, multi-sensor products for surface hailfall diagnosis. *E-Journal of Severe Storms Meteorology*, **13**, 1–36.

Orville, R. E., 2008: Development of the national lightning detection network. *Bulletin of the American Meteorological Society*, **89**, 180 – 190.
URL https://journals.ametsoc.org/view/journals/bams/89/2/bams-89-2-180.xml

Pessi, A. T., S. Businger, K. L. Cummins, N. W. S. Demetriades, M. Murphy, and B. Pifer, 2009: Development of a long-range lightning detection network for the pacific: Construction, calibration, and performance. *Journal of Atmospheric and Oceanic Technology*, **26**, 145 – 166.
URL https://journals.ametsoc.org/view/journals/atot/26/2/2008jtecha1132_1.xml

Picton, P., 1994: *What is a Neural Network?*, Macmillan Education UK, London. 1–12.
URL https://doi.org/10.1007/978-1-349-13530-1_1

Pohjola, H. and A. Mäkelä, 2013: The comparison of gld360 and euclid lightning location systems in europe. *Atmospheric Research*, **123**, 117–128, 6th European Conference on Severe Storms 2011. Palma de Mallorca, Spain.
URL https://www.sciencedirect.com/science/article/pii/S0169809512003456

Potvin, C. K., C. Broyles, P. S. Skinner, H. E. Brooks, and E. Rasmussen, 2019: A bayesian hierarchical modeling framework for correcting reporting bias in the u.s. tornado database. *Weather and Forecasting*, **34**, 15 – 30.
URL https://journals.ametsoc.org/view/journals/wefo/34/1/waf-d-18-0137_1.xml

Potvin, C. K. and M. L. Flora, 2015: Sensitivity of idealized supercell simulations to horizontal grid spacing: Implications for warn-on-forecast. *Monthly Weather Review*, **143**, 2998 – 3024.
URL https://journals.ametsoc.org/view/journals/mwre/143/8/mwr-d-14-00416.1.xml

Púčik, T., P. Groenemeijer, D. Rýva, and M. Kolář, 2015: Proximity soundings of severe and nonsevere thunderstorms in central europe. *Monthly Weather Review*, **143**, 4805 – 4821.
URL https://journals.ametsoc.org/view/journals/mwre/143/12/mwr-d-15-0104.1.xml

Romine, G. S., C. S. Schwartz, J. Berner, K. R. Fossell, C. Snyder, J. L. Anderson, and M. L. Weisman, 2014: Representing forecast error in a convection-permitting ensemble system. *Mon. Wea. Rev.*, **142**, 4519–4541.

Ronneberger, O., P. Fischer, and T. Brox, 2015: U-net: Convolutional networks for biomedical image segmentation.

School of Meteorology/University of Oklahoma, 2021: Gridrad-severe - three-dimensional gridded nexrad wsr-88d radar data for severe events. *Research Data Archive at the National Center for Atmospheric Research, Computational and Information Systems Laboratory*.
URL https://doi.org/10.5065/2B46-1A97

Sha, Y., D. J. Gagne, G. West, and R. Stull, 2020: Deep-learning-based gridded downscaling of surface meteorological variables in complex terrain. part ii: Daily precipitation. *Journal of Applied Meteorology and Climatology*, **59**, 2075 – 2092.
URL https://journals.ametsoc.org/view/journals/apme/59/12/jamc-d-20-0058.1.xml

Smith, B. T., R. L. Thompson, J. S. Grams, C. Broyles, and H. E. Brooks, 2012: Convective modes for significant severe thunderstorms in the contiguous united states. part i: Storm classification and climatology. *Weather and Forecasting*, **27**, 1114 – 1135.
URL https://journals.ametsoc.org/view/journals/wefo/27/5/waf-d-11-00115_1.xml

Smith, T. M., 2014: Performance of a real-time 3dvar analysis system in the hazardous weather testbed. *Wea. Forecasting*, **29**, 63–77.

Stensrud, D. J., L. J. Wicker, M. Xue, D. T. Dawson, N. Yussouf, D. M. Wheatley, T. E. Thompson, N. A. Snook, T. M. Smith, A. D. Schenkman, C. K. Potvin, E. R. Mansell, T. Lei, K. M. Kuhlman, Y. Jung, T. A. Jones, J. Gao, M. C. Coniglio, H. E. Brooks, and K. A. Brewster, 2013: Progress and challenges with warn-on-forecast. *Atmospheric Research*, **123**, 2–16, 6th European Conference on Severe Storms 2011. Palma de Mallorca, Spain.
URL https://www.sciencedirect.com/science/article/pii/S016980951200110X

Stensrud, D. J., M. Xue, L. J. Wicker, K. E. Kelleher, M. P. Foster, J. T. Schaefer, R. S. Schneider, S. G. Benjamin, S. S. Weygandt, J. T. Ferree, and J. P. Tuell, 2009: Convective-scale warn-on-forecast system: A vision for 2020. *Bulletin of the American Meteorological Society*, **90**, 1487 – 1500.
URL https://journals.ametsoc.org/view/journals/bams/90/10/2009bams2795_1.xml

Stone, M., 1974: Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, **36**, 111–133.
URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1974.tb00994.x

Thompson, R. L., R. Edwards, J. A. Hart, K. L. Elmore, and P. Markowski, 2003: Close proximity soundings within supercell environments obtained from the rapid update cycle. *Weather and Forecasting*, **18**, 1243 – 1261.
URL https://journals.ametsoc.org/view/journals/wefo/18/6/1520-0434_2003_018_1243_cpswse_2_0_co_2.xml

Tuovinen, J.-P., J. Rauhala, and D. M. Schultz, 2015: Significant-hail-producing storms in finland: Convective-storm environment and mode. *Weather and Forecasting*, **30**, 1064 – 1076.
URL https://journals.ametsoc.org/view/journals/wefo/30/4/waf-d-14-00159_1.xml

Vaisala, 2023: Lightning strike detector.
URL https://www.vaisala.com/en/digital-and-data-services/lightning

Wang, H. and B. Raj, 2017: On the origin of deep learning on the origin of deep learning.

Wendt, N. A. and I. L. Jirak, 2021: An hourly climatology of operational mrms mesh-diagnosed severe and significant hail with comparisons to storm data hail reports. *Weather and Forecasting*, **36**, 645 – 659.
URL https://journals.ametsoc.org/view/journals/wefo/36/2/WAF-D-20-0158.1.xml

Witt, A., M. D. Eilts, G. J. Stumpf, J. T. Johnson, E. D. W. Mitchell, and K. W. Thomas, 1998: An enhanced hail detection algorithm for the wsr-88d. *Weather and Forecasting*, **13**, 286 – 303.
URL https://journals.ametsoc.org/view/journals/wefo/13/2/1520-0434_1998_013_0286_aehdaf_2_0_co_2.xml

Yano, J.-I., M. Z. Ziemiański, M. Cullen, P. Termonia, J. Onvlee, L. Bengtsson, A. Carrassi, R. Davy, A. Deluca, S. L. Gray, V. Homar, M. Köhler, S. Krichak, S. Michaelides, V. T. J. Phillips, P. M. M. Soares, and A. A. Wyszogrodzki, 2018: Scientific challenges of convective-scale numerical weather prediction. *Bulletin of the American Meteorological Society*, **99**, 699 – 710.
URL https://journals.ametsoc.org/view/journals/bams/99/4/bams-d-17-0125.1.xml

Özgün Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, 2016: 3d u-net: Learning dense volumetric segmentation from sparse annotation.

# Appendices

## A.1 Further Data Distribution Discussion

The convection performance/hail rarity compound issue, when paired with haphazardous data distribution, was found to be the primary source of our limited performance in the early stages of this thesis. Addressing the issue led to the largest single increase in performance across all non-lightning experiments performed in this thesis. It is critical to note that the train, validation, and test set delineations were entirely based upon dates within our dataset date range for all discussed experiments performed up until this stage. This had significant implications for introducing differing biases into our train, validation, and test sets. From examining the number of samples produced by various date delineation combinations, we settled on: 2017-01-01 to 2018-05-10 for our train set, 2018-05-11 to 2018-12-31 for validation set, and 2019-01-01 to 2019-12-31 for our test set (all of these dates outlined in Table A.1). However, it should be noted that only some of the days within these ranges contained data as part of our WoFS and GridRad datasets. Therefore, the beginning and final days containing data were not necessarily the beginning and end dates given in these ranges.

Using dates as a method to split a dataset into train/validation/testing partitions is common throughout the meteorological machine learning community. It has several advantages including but not limited to representing a test dataset as an unseen dataset from a new operational year. However, as noted above, the convection performance/hail rarity compound issue present in this thesis rendered any unshuffled data as detrimental to performance quality. The smaller size of our dataset resulted in overly frequent cases of overfitting to the validation set, while any test set analysis yielded results of far poorer quality to the validation set due to the WoFS convection-related variance between them (see Figure 4.1). The analysis highlighted by these figures was the reason for our use of composite reflectivity observations supplied by the GridRad

**Dataset Distribution History**

| | Train | Validation | Test |
|---|---|---|---|
| Early Experiments (Quality Issues) | 2017-01-01 to 2018-05-10 | 2018-05-11 to 2018-12-31 | 2019-01-01 to 2019-12-31 |
| Intermediate Experiments | 2017-01-01 to 2018-12-31 | | 2019-01-01 to 2019-12-31 |
| Made for Shuffle-Proving Figure | 2017-01-01 to 2021-04-26 | | 2021-04-27 to 2021-12-31 |
| Final Experiments (Results Chapter) | 2017-01-01 to 2021-12-31 | | |

= Clustering-Based Shuffling  = No Shuffling

Table A.1: A table of the dates used for all dataset partitions. Cells highlighted in green indicate data drawn from the indicated dates either randomly or from using cross-validation, but always with the storm clustering system. Cells highlighted in blue indicate partitions sliced by the given dates with no additional cross-date shuffling.

dataset introduced in section 3.1.3. These convection quality figures were a simple critical success index time series created from the direct pixelwise comparison of WoFS composite reflectivity greater than 40 dBZ and GridRad composite reflectivity observations greater than 40 dBZ. The critical success index was calculated for the batch of all samples within each dataset at each timestep. The 40 dBZ threshold was chosen to represent convection as this is a commonly accepted threshold in the convective weather community.

After the WoFS convection performance discovery was made during our early experiments, it was decided that this compound issue was most logically resolved with performing robust cross-validation. Cross-validation has been shown to be a viable solution to problems induced by a small/rare/unstable dataset (Allen 1974). This was because cross-validation is designed to introduce multiple levels of shuffling while also ensuring multiple validation sets that would allow for varying subsets of the rare-label containing data. The shuffling would in theory help to produce train and validation

sets that would better represent the natural distribution of hail. While multiple validation sets were expected to help avoid overfitting validation which is a known problem with smaller datasets.

The cross-validation process used for the remainder of this thesis was outlined in more detail in sections 3.2.2.1 and 3.2.2.3. Stratified group five-fold cross-validation was our selected method. Grouping was required because samples could not simply be randomly pulled from the complete dataset as many samples were from the same storm system on the same calendar day. As discussed in section 3.2.2.1 the grouping method used was based upon "storm days". Each "storm day" was defined as any number of samples with less than 6 hours of time between each sample initialization time. These "storm days" were all considered individual inseparable units that the cross-validation algorithm had to sort in either the train or validation sets with each split. In summary, no two samples from the same "storm day" were allowed to exist in both the train and validation sets. Furthermore, a stratified cross-validation algorithm was selected because stratification enforced similar distributions of hail labels for each train/validation pair which was particularly valuable for a rare label dataset.

At this intermediate stage of the thesis this cross-validation algorithm produced a shuffled set of 5 train and validation pairs, however, it was decided to continue to use data from 2019-01-01 to 2019-12-31 as our set aside test set. The objective of this was to gain the advantages of the cross-validation without losing the aforementioned advantages posed by keeping the test set delineated by a calendar date. In summary, experiments performed at this stage were on machine learning models trained and validated on cross-validation sourced shuffled data from 2017-2018. The test data was made up of "storm days" only from 2019 and had no earlier data shuffled within. Unfortunately, limited improvements were again observed in the test results. The source of this result was believed to have originated from the large differences in convection

forecast performance between the delineated sets as shown in Figure 4.1. It was immediately clear that the test set would also have to be from a shuffled source.

This last shuffling subsequently became the final adjustment made to the data engineering process which culminated in the solution to the convection performance/hail rarity compound issue. As shown in Figure 3.4 once patches had been created from all available data sources for all available dates (for this experiment 2017-2019), 20% of this complete dataset was randomly sampled and set aside for the test set. Although randomness was essential to maintain the integrity of the test set, "storm day" clustering and subsequent grouping was required again to ensure no two samples from the same "storm day" could exist in both the test set and remaining data. After the test set was set aside, the remaining data was again used in the stratified group five-fold cross-validation algorithm. In summary, a total of 5 train sets, 5 validation sets, and 1 now randomly-sampled test set were generated. As shown in 4.1 a substantial improvement in test set performance was observed. A general discussion of practices pertaining to dataset quality and setup is likely grounds for an entirely independent study. It was clear from the results of this experiment that any aspiring machine learning meteorologist could benefit considerably from preemptive knowledge regarding his or her dataset quality and any subsequent recommended dataset configurations.

In conclusion, we settled on the aforementioned 20% randomly sampled partition of our test set. The remaining data was then partitioned into 5 cross-validation splits for a total of 5 train sets and 5 validation sets as described in Figure 3.4. However, in the final stages of this thesis, two additional years worth of GridRad MESH data was produced for our use. For these two years, the GridRad MESH data was our only absent data, thus with this addition we were able to add two more years worth of patches. Therefore, in these final stages (all primary experiments discussed in results) our data consisted of the years 2017-2021 (as opposed to just 2017-2019). The aforementioned

5 shuffled train sets, 5 shuffled validation sets, and the single shuffled test set used for all remaining experiments were all created from these years.

## A.2 Early Experiments

The many other experiments performed during the two years spent on this project also necessitated some amount of detailed discussion. With each experiment performed, varying degrees of exhaustive machine learning optimization methods were used. We described this process as having varying degrees of optimization because at the later stages of the project higher quality and more rigorous techniques were used. However, computationally expensive and time-consuming techniques such as hyperparameter searches were performed throughout the duration of the project. These techniques required the training of numerous individual machine learning models. This resulted in the training of hundreds of individual models with every distinct experiment performed. Over the entirety of the project, this resulted in the creation of thousands of individual machine learning models (each with expensive computation costs).

Almost all of these experiments were built around the manipulation of the datasets used for the training of all the machine learning models. This dominant theme throughout the project coincides with the traditional machine learning concept that machine learning work consists mostly of data engineering. We have ordered these secondary experiment results from least significant to more significant. Firstly, several experiments were performed on the scale and sampling distributions of the patches used for training the models. Several different square patch sizes were tested for their effect on test set performance in the early stages of the project. By the conclusion of the project, patch sizes of 32x32, 64x64, and 128x128 were tested (powers of 2 were utilized since downscaling and upscaling operations inside U-Nets were easier when each layer could

be divided by 2). These sizes were tested simultaneously to several of the other experiments discussed in this section (and with exhaustive hyperapter searches), however, the effect of the size changes themselves could still be isolated. The expected behavior of these size changes was that smaller patches would enable a greater number of samples, while larger sizes would allow for patches that could resolve more structure. Traditional machine learning knowledge dictated that we should attempt to have datasets that are as large as possible and we expected larger patches to perform segmentation better due to a greater capture of storm structure. Thus, a balancing problem was expected between these sizes where neither patches that were too small or too large were acceptable.

However, it was noted throughout these experiments that these three sizes had little effect on the final summary test performance. It was believed that this result occurred because the effects caused by the extreme rarity of hail events and the limited size of our dataset dominated the experiments (specifically the number of pixels containing hail labels was exceedingly small for a given dataset). Following these experiments all future models were trained on 64x64 sized patches as this size seemed to be a sufficient compromise for the size balancing problem. The most important outcome of these early sizing experiments was that we determined the most significant problem that had to be overcome by this whole project was the rare hail label issue. Therefore, most subsequent experiments were performed with the objective of mitigating this issue.

The experiments that were performed immediately following the sizing experiment were the aforementioned (earlier in this section) patch sampling distribution experiments. The motives behind these experiments were to maintain the large patch structuring advantage that were obtained from 64x64 and 128x128 sized patches while vastly increasing the dataset size to offset the hail's rarity. Thus far in the project, we exclusively set our AutoPatcher to slice patches with a uniform distribution, where the

maximum number of patches that could fit starting from the upper left corner of the domain were used. Instead, to increase the number of patches that could be pulled from a particular domain, AutoPatcher was set to create patches with overlapping pixels. Still however, no exactly identical patches were allowed to enforce some degree of the traditional idea of independent and identically distributed (IID) data. The objective behind this system was to create an effect similar to the process of "image augmentation" used in other image-based machine learning applications. For example, we were aiming to allow the same storm to exist in multiple patches (or samples) but in different positions within the patch bounds. This was expected to allow the machine learning model to treat each of these translated versions of the same storm as new storms and thus increase the dataset size and subsequent performance. It was also anticipated that this would reduce spatial biases that could arise if some storms consistently appeared in the same place several times.

Unfortunately, this too did not increase performance to a significant degree. Thus, we continued on with experiments that aimed to address the rare hail label issue. One option for an experiment that was anticipated to alleviate this issue was an attempt to enforce a form of class balancing among samples (patches) within the training dataset. This was achieved by setting AutoPatcher to produce an even number of no-hail-containing and hail-containing patches. We defined no-hail-containing patches as those with no hail pixels and hail-containing patches as those with at least n hail pixels where n was a defined hyperparameter. Traditionally, this method was expected to alleviate the rare hail label issue by artificially inflating the number of hail-containing patches available to the model during train time as opposed to the rare natural distribution of hail pixels if we were to sample randomly. This experiment also failed to produce significant results. We believe this was because, when still allowing overlap, changing the sampling method in this manner produced the same conceptual outcome as what

occurred in the overlap experiment discussed previously. Specifically, changing the frequency of patches that contained hail data did not have a strong enough effect on performance when the total number of hail events (and dataset size) remained fixed and relatively small. Likewise, when this experiment was performed while not allowing the patch overlap mode, no substantial test performance increases were noted. In that case, it was believed that the vastly reduced dataset size was the primary source of failure as requiring a balanced number of hail and no-hail patches necessitated a greatly reduced number of no-hail patches when no pixels were allowed to be repeated.

Following these experiments, it was decided that the next logical attempt at addressing the rare hail label issue was centered around the internal architectures of the machine learning models themselves. It was hypothesized that some form of neighborhooding would improve test performance since the spatial and temporal density of hail labels was inherently sparse with the rare hail label issue. At a high level, neighborhooding was expected to allow some spatial tolerance (and temporal tolerance for architecture experiment 3) for when prediction pixels were offset from observed hail label pixels. The most direct way to incorporate neighborhooding was to introduce a neighborhood-based loss to our model architectures. We elected to use fractions skill score loss (FSS) for our experiments as it has been shown to be successfully implemented in several meteorological machine learning studies. These experiments also failed to produce meaningful test results and it was even found that using FSS loss greatly increased the probability that no optimized machine learning model could be found during hyperparameter searches. It was believed that this optimization problem arose jointly because of two issues. Firstly, the FSS loss vastly increased the complexity of the machine learning model that had to be trained which limited an optimizer's ability to avoid saddle points. It also arose because the general sensitivity to spatial scaling of storm features with a mesoscale problem limited the success of a spatially

smoothing loss like FSS which was expected to disregard some of this spatial structure. However, there are many different approaches to neighborhooding that are used throughout meteorology. One such method was to expand training (and sometimes validation but never test) labels with either an image filter or manually pixelwise while keeping the loss pixel-based. Additionally, metrics that exploited neighborhooding had considerable value as these could be used post-training with no effect on training performance and for little additional computation cost. These were used extensively in our result figures.

The final set of experiments that yielded few substantial results were based around the manipulation and selection of predictor features. These experiments had less of an expected impact on the rare hail label issue, however it was still deemed important to investigate the effect of certain features on test performance. In the early stages of this thesis, WoFS predictors were removed and introduced with limited changes in performance. This was believed to be the case because many WoFS predictors came from similar groupings in meteorological terms (for example multiple WoFS predictors can represent wind shear) and as such removing entire groups was required to induce performance changes. What was expected to be the most significant experiment in this category was the adding and removal of WoFS hail fields such as HAILCAST (Adams-Selin and Ziegler 2016). It was believed that the best surrogate for hail present in WoFS would have a meaningful effect on our own machine learning forecast of hail when this surrogate was used as a predictor. However, the manipulation of these predictors likewise produced little change in test performance. It was hypothesized that both the manipulation of non-hail and hail WoFS predictors produced little changes in test performance because the success of WoFS fields as predictors is primarily based upon the ability of WoFS to successfully forecast convective initiation and development.

## A.3 AutoPatcher Details

The process used by AutoPatcher is outlined in figure A.1 and each step is described in this section with reference to this figure. AutoPatcher is designed to take any number of netcdf datasets with varying domains and fit them together in time and space. It required one top level configuration file that controlled the main settings of the system, and a set of configuration files for the datasets (one for each dataset such as WoFs, GridRad, etc.). Configuration files were needed for each dataset so that each dataset's unique characteristics could be defined independently.

The data storage directory structures can vary between datasets so regular expressions were used to define file positions and file naming conventions. The backbone of the logic that drives AutoPatcher is its use and manipulation of a dataset's time dimension (steps 1,2,3,and 4 in Figure A.1). Some datasets only have their data spaced over real time, while others (specifically data from NWP) also have initialization times that must be considered. In most earth science conventions time is either represented by something within the file names (often when each .nc file corresponds to each time step) or time is represented as a dimension within the file itself. Both cases had to be handled to robustly cover the range of datasets that were needed for this thesis. For the "time in filename" case, regular expressions were again used to denote the time's positioning within the string of the file name and to describe its ISO date convention. Time could also be extracted from any directory structure names used in the tree of directories above each file using an array of regular expressions. For example: a file labeled "2005.nc" (hour and minute) could be nested in a directory labeled "0802" (month and day) which in turn could be nested in "2005" (year). Any combination of these in any position had to be defined entirely within the configuration files for each dataset to again avoid hardcoding. For the "time inside file" case, time data contained

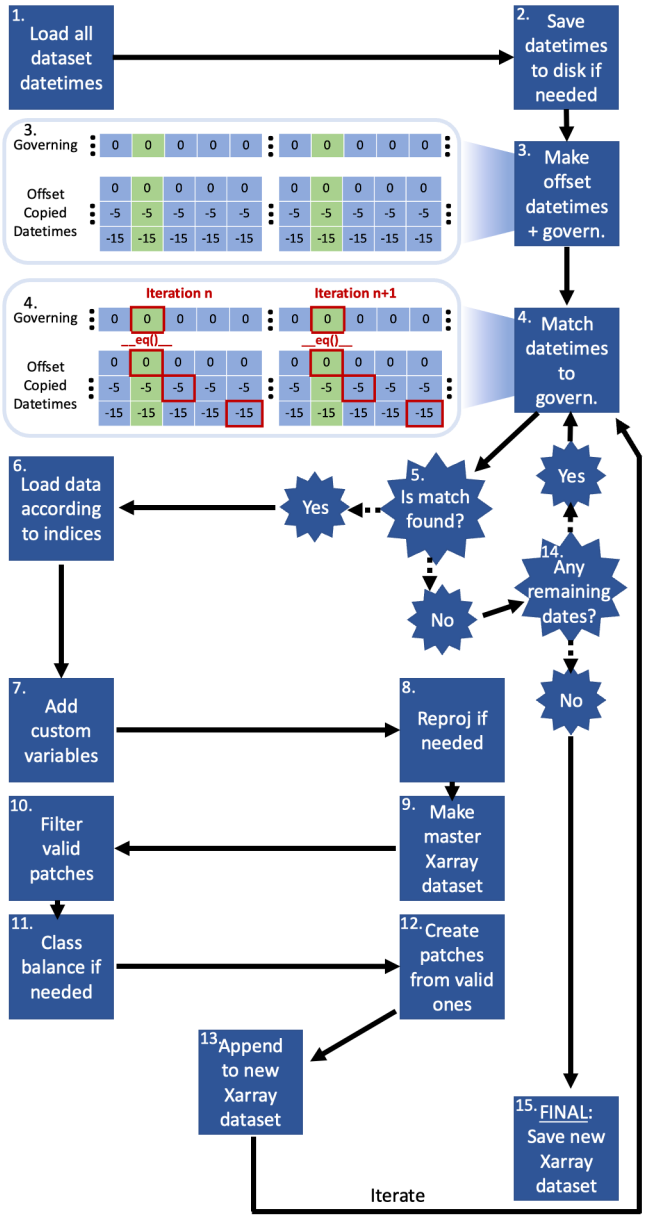



Figure A.1: A flowchart of the autopatcher process. Each square is a primary software step. The green colors in each of the expanded sections indicate timesteps that are NWP initialization times. Number in that section indicates how many minutes the datasets are offset by to make the matches. The red boxes indicate successful matches. Each star is a conditional logic step. All steps are numbered and are accordingly explained throughout 3.2.1

internally was preemptively loaded and saved to disk to avoid flooding the disk with too many accesses during the software's search phase (step 2 in Figure A.1).

This search phase is when AutoPatcher uses a combination of iterative logic and vectorized programming to comb through the given datasets to find examples where all datasets concurrently line up in space and time (step 1 in Figure A.1). This worked by first labelling a single dataset as the "governing dataset". This dataset's time data was then iteratively used to compare like times across all datasets with numpy vectorization. This process was chosen as it exploits numpy's efficient use of homogeneous arrays and C code to vastly reduce runtimes for the large datasets we worked with. This system also had to support the finding of multiple timesteps from the same dataset for a particular sample because we needed the ability to have a range of predictors across time to support more complex forecasting techniques. This was achieved by shifting each dataset's full set of datetimes by the number of minutes needed for each feature's time step and then re-adding the set to the full datetime array (step 3 in Figure A.1). The result was that when vectorized numpy searches were performed datetimes were matched together from the desired predefined offset times (step 4 in Figure A.1). This created datasets with multiple timesteps neatly combined together in an additional time dimension. In order to encapsulate the initialization time logic needed for NWP datasets, different objects would have to be created in place of numpy's datetime64 and timedelta64 objects. This worked by overriding the __eq__() python method contained within the objects so they will be compatible with numpy vectorized searching. We called these new objects DatetimePair and TimedeltaPair. Two DatetimePairs were considered equal (and thus matched in the searching process) if both their initialization times (if dataset used initialization time) and their valid times matched. This all came together to produce a fast and efficient way to merge many thousands of samples together across any n number of differing datasets. This whole process is visualized in

the two expanded sections of step 2 and 3 in Figure A.1. Following this if a match is found in this "numpy equals-operator" process, data is loaded into memory according to to the indices of the found timesteps (step 6 in Figure A.1).

As mentioned earlier in this section, varying spatial domains and resolutions also posed a significant hurdle that had to be overcome. AutoPatcher also had to be designed to take this issue into account robustly for any dataset. This was handled by using the bilinear reprojection method employed by the xESMF (Earth System Modelling Framework) Universal Regridder for Geospatial Data python package (used in step 8 in Figure A.1). Within each dataset configuration file, the user can designate whether a dataset needs reprojection and whether its domain would be a target for reprojection or whether it would receive the reprojection. For the purposes of this project, we reprojected both the GridRad dataset (1 km resolution, varying but larger domain) and the Vaisala lightning dataset (same as GridRad) onto each WoFS domain (3 km smaller domain). This direction was chosen because it was more accurate to down-sample from high resolution data onto lower resolution data. Additionally, the WoFS domains were considerably smaller than the other datasets' domains and moved all over the the other domains since WoFS domains moved with severe weather events. So it made more sense to project onto these smaller domains and save a longer computation time that would occur with the larger domains. Once all datasets were loaded using their spatial and temporal information, they were combined into one single large Xarray dataset object that could define all the dimension information of each dataset without the need for costly array broadcasting (step 9 in Figure A.1). This "master" Xarray dataset then formed the backbone of the rest of the AutoPatcher process as its easy to use format allowed for high-level modifications in every subsequent method.

These subsequent functions included but were not limited to: the ability to select any desired data variables (examples include composite reflectivity from WoFS

or MESH from GridRad), control of any extra dimensions (for example a ensemble member dimension from a NWP dataset), creation of any class labels, the applying of a modifying function to any variable, filtering of any data, class balancing, and patch ordering (steps 7, 10, and 11 in Figure A.1). Exploiting the robust methods included within Xarray dataset objects such as slicing allowed us to add the data select and dimension control sections. While class labels, modifying functions, and filtering were all handled by user-given custom functions that produced some result given the "master" dataset. After these modifications were applied, any remaining valid patches were sliced from the master Xarray dataset and were then appended to a new Xarray dataset that would be used for output (steps 12 and 13 in Figure A.1). This process was then iterated to find further initialization times until all data was exhausted and the new Xarray dataset could be outputted to disk (steps 14 and 15 in Figure A.1).

The end result of this whole system was a piece of software that could generate patches from any number of dataset. It also allowed for easy tuning of data-specific hyperparameters throughout the project. This resulted in a large range of machine learning models generated and tested by the project's conclusion.

## A.4 Used Hyperparameters

**Hyperparameters for Final Version of Experiments**

| | Convolutional Layers | Kernel Size | Activation | Num of Kernels | Depth | Optimizer | Batch Norm | 3-Plus | Batch Size | Learning Rate | L2 Regularization | L1 Regularization |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Exp. 3 (3D UNet) | 2 | 5x5 | ReLU | 8 | 3 | SGD | Yes | Yes | 32 | 0.001 | 0.01 | 0.001 |
| Exp. 2 (15 mins) | 1 | 7x7 | ReLU | 16 | 3 | SGD | Yes | Yes | 32 | 0.0001 | 0.01 | 0.00001 |
| 20 mins | 3 | 7x7 | ReLU | 32 | 2 | SGD | Yes | No | 256 | 0.001 | 0.1 | 0.005 |
| 25 mins | 2 | 3x3 | ELU | 8 | 1 | Adagrad | Yes | No | 64 | 0.001 | 0.005 | 0.005 |
| 30 mins | 3 | 7x7 | ReLU | 32 | 2 | SGD | Yes | No | 256 | 0.001 | 0.005 | 0.00001 |
| 35 mins | 2 | 5x5 | ReLU | 8 | 2 | SGD | Yes | No | 32 | 0.01 | 0.00001 | 0.001 |
| 40 mins | 2 | 5x5 | ReLU | 8 | 2 | SGD | Yes | No | 32 | 0.01 | 0.00001 | 0.001 |
| 45 mins | 2 | 5x5 | ReLU | 8 | 3 | SGD | Yes | Yes | 32 | 0.001 | 0.01 | 0.001 |
| 50 mins | 2 | 5x5 | ReLU | 8 | 3 | SGD | Yes | Yes | 32 | 0.001 | 0.01 | 0.001 |
| 55 mins | 2 | 5x5 | ELU | 16 | 3 | Adagrad | No | No | 32 | 0.01 | 0.01 | 0.00001 |
| Exp. 1 | 1 | 5x5 | ELU | 4 | 3 | SGD | Yes | No | 32 | 0.001 | 0.05 | 0.01 |

Table A.2: A table of all hyperparameters used for the optimized U-Nets in all 3 architecture experiments. All architecture experiments are delineated by the white spacing. Architecture experiment 3 is the top, architecture experiment 2 is in the middle, and architecture experiment 1 is at the bottom.