DISSERTATION

# FROM NEURO–INSPIRED ATTENTION METHODS TO GENERATIVE DIFFUSION: APPLICATIONS TO WEATHER AND CLIMATE

Submitted by

Jason Stock

Department of Computer Science

In partial fulfillment of the requirements

for the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Fall 2024

Doctoral Committee:

Advisor: Chuck Anderson

Imme Ebert-Uphoff
Nikhil Krishnaswamy
Sarath Sreedharan

# ABSTRACT

## FROM NEURO–INSPIRED ATTENTION METHODS TO GENERATIVE DIFFUSION: APPLICATIONS TO WEATHER AND CLIMATE

Machine learning presents new opportunities for addressing the complexities of atmospheric science, where high-dimensional, sparse, and variable data challenge traditional methods. This dissertation introduces a range of algorithms, motivated specifically by the intricacies of weather and climate applications. These challenges complement those that are fundamental in machine learning, such as extracting relevant features, generating high-quality imagery, and providing interpretable model predictions.

To this end, we propose methods to integrate adaptive wavelets and spatial attention into neural networks, showing improvements on tasks with limited data. We design a memory-based model of sequential attention to expressively contextualize a subset of image regions. Additionally, we explore transformer models for image translation, with an emphasis on explainability, that overcome the limitations of convolutional networks. Lastly, we discover meaningful long-range dynamics in oscillatory data from an autoregressive generative diffusion model—a very different approach from the current physics-based models. These methods collectively improve predictive performance and deepen our understanding of both the underlying algorithmic and physical processes.

The generality of most of these methods is demonstrated on synthetic data and classical vision tasks, but we place a particular emphasis on their impact in weather and climate modeling. Some notable examples include an application to estimate synthetic radar from satellite imagery, predicting the intensity of tropical cyclones, and modeling global climate variability from observational data for intraseasonal predictability. These approaches, however, are flexible and hold potential for adaptation across various application domains and data modalities.

# ACKNOWLEDGEMENTS

I wish to thank those who have helped to support me throughout my doctoral studies. I am particularly indebted to my advisor, Chuck Anderson, who has guided me since my undergraduate days. You always remind me of one of the greatest scientific lessons—to stay curious—while consistently encouraging me to grow and showing me simplicity in even the most complex of problems. I am infinitely grateful and fortunate to have had you as my advisor.

Much of this dissertation would not have been possible without the guidance of Imme Ebert-Uphoff. You helped me realize the potential of interdisciplinary research by continually providing invaluable feedback, offering lasting connections, and being enthusiastic about new ideas, all of which were integral to my success.

I would also like to thank my other committee members, Nikhil Krishnaswamy and Sarath Sreedharan, on their feedback through multiple iterations of this dissertation. Similarly, I wish to thank my mentors Andrew Fagg, Mike Pritchard, Jaideep Pathak, and Bryan Doyle for the thought-provoking discussions and challenging my perspectives for the better.

To many of those closest to me, I appreciate your patience and understanding over the years, particularly, my mother, father, and brothers. You all helped me reconnect with the world and grow my fundamental interests, while allowing and reminding me to stay academically driven. To my dearest friends, Tom, Parker, and Waylon, I am grateful for the consistent motivation, support, and help in keeping me grounded. Also, if it were not for our feline friends, I feel the field of machine learning would be arguably less enticing, so I am grateful for all the kitties I am surrounded by.

Importantly, thank you, Stephanie, for being a constant source of light and my greatest supporter. Your unwavering belief in me and your presence throughout the challenges I faced made this journey not only possible but meaningful, no matter where in this world we were.

This dissertation was largely supported by NSF Grant No. 2019758, AI Institute for Research on Trustworthy AI in Weather, Climate, and Coastal Oceanography (AI2ES). Beyond graciously funding my academics, the connections and support from AI2ES fostered formative learning experiences and professional development opportunities that will carry with me. Additionally, part of this work was supported by NVIDIA during an internship with the Climate Simulation Research Group, for which I am incredibly grateful to have had.

TABLE OF CONTENTS

# Chapter 1

# Introduction

The application of machine learning for Earth science is a treacherous landscape to navigate. The challenges in both computer and atmospheric science, more specifically, are distinct. However, they often complement each other, creating a continuous interplay where solving one problem reveals new challenges or even opportunities in the other. This interplay, and the result of interdisciplinary efforts thereof, drives scientific discovery and the demand for novel solutions that simultaneously push the boundaries of each field. The goal of this dissertation is to derive a set of machine learning algorithms that go beyond traditional designs, not only addressing these challenges but also elucidating the appropriate use of methods for specific applications.

In machine learning, and in computer vision more specifically, there are a number of significant challenges. One of the foremost difficulties lies in extracting meaningful and salient feature representations from complex, high-dimensional data. These features are crucial for model performance and derived explanations, yet models such as convolutional and transformer-based networks often rely on abstract representations that lack structurally innate patterns and domain-specific insights. Despite advances in model architectures, this challenge is amplified when tasks have relatively few training samples to learn from [1, 2]—a common scenario when modeling direct, sparse, and spatiotemporal observations of our world.

There are also several challenges and observations when assessing the output of these models, of which we emphasize two. The first is understanding how a model arrives at its predictions and with what features from the input were used to form a prediction—part of a larger field known as explainable artificial intelligence [3]. The second, related to image-to-image translation, involves generating sharp and accurate estimates [4, 5]. In this context, sharpness refers to the clarity of features, with well-defined edges and realistic features. Addressing both of these challenges is essential for improving performance and deepening our understanding of machine learning.

Improvements to generating sharp and accurate imagery have been made with the advent of generative models [6, 7]. Traditional, discriminative models focus on learning direct input-output mappings, whereas generative models aim to capture the underlying data distribution. This allows us to sample from the distribution, often conditionally, to generate an ensemble or variety of outputs. While the network architectures

can be similar in both cases, generative models differ in their training objectives and sampling processes. However, when applied autoregressively to generate sequences, the long-term dynamics of these models remain poorly understood. This is particularly relevant in atmospheric science, where small errors in chaotic spatial environments can accumulate over time.

The challenges within atmospheric science, particularly in weather and climate modeling, are multifaceted but often complement those discussed above. Foremost, gridded atmospheric data often consists of high-dimensional, multi-channel features that represent a range of physical properties, from observable radiances to numerical analyses of assimilated observations. The relationships between channels are evidently more meaningful than natural red-green-blue imagery. Moreover, spatiotemporal observational data from satellites or ground-based instruments are frequently sparse due to inherent data biases, geographical influences, or the disproportional occurrence of atmospheric phenomena [8–10]. This can lead to smaller datasets or incomplete (or biased) observations.

The dynamics of the atmosphere and its forcings are also highly nonlinear and variable in nature. This inherent variability makes reliable long-range forecasts particularly difficult, as numerical weather prediction and climate models have different assumptions and numerical uncertainty that can lead to biases [11, 12]. Beyond forecasting, the atmospheric variability itself is difficult to model physically and is still incompletely understood [13, 14]. This complexity poses challenges for accurately representing tropical dynamics and predicting the intensity and development of severe weather systems. Moreover, the structural scale varies widely, with some phenomena driven by global climate forcings, while others, like convective storms, are more localized.

These challenges underpin the need for methods that can generalize, often in the overparameterized regime, while realistically capturing internal variability and considering the varying scales of atmospheric events. Furthermore, there is a pressing need for such methods to be trustworthy and reliable, especially in high-stakes environments where predictions directly impact decision-making [8]. Considerable progress has been made through convergent interdisciplinary research, integrating the fields of machine learning, atmospheric science, and risk communication [15, 16]. However, there is a continual need for this kind of research to further drive the field and address these challenges.

A dense body of prior and concurrent work has gone into addressing some of these challenges, and effectively forming the basis and motivation for this dissertation. This includes classical mathematical approaches for feature extraction [17, 18] to large-scale data-driven models. The latter comprise short-

to medium-range global weather forecasting models [19–24], which are approaching the performance of numerical weather prediction at a fraction of the computational cost. Moving from weather to climate timescales, machine learning has also been used to identify indicators of climate change [25–28] and in assessing seasonal variability [29, 30]. In the chapters that follow, we will explore these works in more detail, but provide the context for our research objectives presented here.

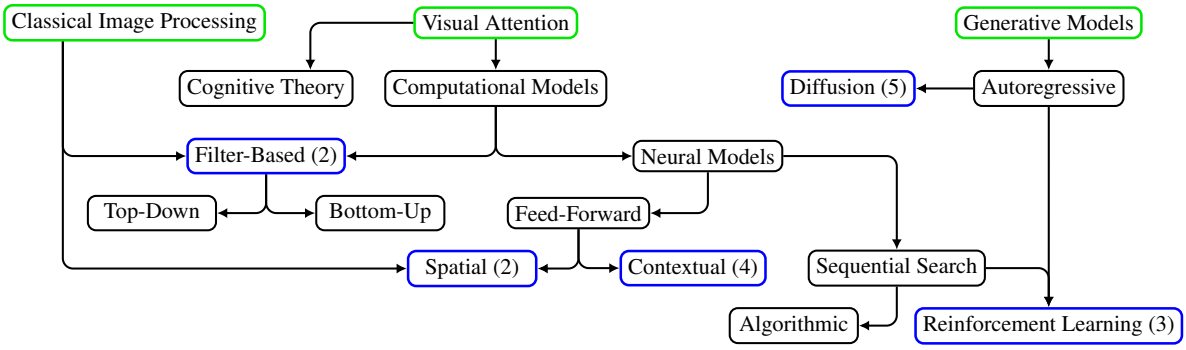



**Figure 1.1:** Categorical overview and relationship of the concepts discussed herein. In green are the top-level categories of well-studied research areas, and in blue (with chapter numbers) are the sub-fields that are a particular focus in this dissertation.

## 1.1 Research Objective and Methods

There is no "one-size-fits-all" algorithm to solve the problems in atmospheric science, and this dissertation does not aim to convince the reader otherwise. Moreover, many of the traditional "out-of-the-box" algorithms can struggle with weather and climate applications, partly due to the aforementioned challenges, e.g., small sample size of labeled observations and the fact that many events of interest, such as hurricanes, are rare. Therefore, we propose focusing on the properties of data that are specific to an application and how new algorithms (i.e., neural network architectures and training objectives) can be designed to conform to these needs. Subsequently, we ask the inverse: what are the capacities and limitations of a particular method for a given application and how does it compare to traditional approaches? The research objective of this dissertation is therefore to develop and evaluate new machine learning algorithms tailored to atmospheric science data, with the dual goal of improving performance and deepening our understanding of these methods and the data itself. By focusing on the design, capabilities, and limitations of these approaches, we aim to

bridge the gap between fundamental model development and practical applications in weather and climate science.

The methods and research areas explored herein are categorized in Figure 1.1, which outlines the general relationships between relevant sub-fields. Each chapter explores these areas in greater depth, focusing on the connections between visual attention, classical image processing, and generative models. These top-level categories encompass a range of approaches, from filter-based methods to reinforcement learning, and this diagram serves as a roadmap for understanding the relationships between the different methods.

**Table 1.1:** A comparison of proposed methods. Note: deterministic (Det.) and probabilistic (Prob.)

| (Ch.) Method | Primary Technique | Climate (✓) Weather (✗) | Det. (✓) Prob. (✗) | Sample Efficiency | Potential Explainability | Computational Complexity |
|---|---|---|---|---|---|---|
| (2) WaveNet | Wavelet Transform | ✗ Gravity Waves | ✓ | low | high | low |
| (2) Scattering | Scattering Transform / Spatial Attention | ✗ Tropical Cyclones / Lightning Prediction | ✓ | medium | medium | medium |
| (3) M-SAtt | Sequential Attention / Reinforcement Learning | ✓ Climate Indicators | ✗ | medium | high | high |
| (4) SRViT | Vision Transformer | ✗ Radar Estimation | ✓ | high | medium | high |
| (5) DiffObs | Diffusion Model | ✓ S2S Prediction | ✗ | high | low | very high |

The first top-level category, *visual attention*, draws from cognitive psychology and neuroscience, and is broken into computational models that include both traditional, filter-based, and neural approaches. Filter-based methods focus on direct feature extraction, whereas neural methods, separated further by feed-forward and sequential search, dynamically learn these features from the data. Spatial attention, a feed-forward method, learns to focus on salient data features, which, along with the filter-based methods, relate to the second top-level category, *classical image processing*. In contrast, contextual attention centers on learning relationships between higher-level features, forming the basis for sequential search methods that build feature sets (or trajectories) over time. While algorithmic solutions exist to optimize these searches, they are framed here within the context of reinforcement learning. Lastly, *generative models* are studied autoregressively, connecting both reinforcement learning and diffusion models in their abilities to model tasks probabilistically.

In this dissertation, we present five methods derived from the aforementioned categories, as summarized in Table 1.1. This table lists each method with its corresponding chapter reference and primary techniques. The design of several methods are inspired by both the data and the application, often with feedback in collaboration with domain scientists (see below). The relationship to their respective application domains,

including whether it is weather or climate science related, are also listed. While most methods are evaluated using observational data, we also demonstrate their efficacy on synthetic data and classical computer vision tasks.

It is important to note that not every method is intended to advance state-of-the-art performance for these application domains. Chapters 2 and 3 are largely demonstrative and did not result from extensive interdisciplinary collaborations. These chapters include applications to illustrate the potential of our methods compared to traditional baselines. By contrast, Chapters 4 and 5 came about through collaboration with domain experts to achieve state-of-the-art (at the time of publication) and support scientific discovery.

This table also compares the computational factors that are *relative* between methods, including: **(a)** *sample efficiency*: ability to generalize proportionally to the dataset size (low indicates limited improvements with more data, and high indicates a joint performance benefit); **(b)** *potential explainability*: accessibility to methods, inherent or not, for interpreting model predictions (low indicates more challenging, and complex methods, while high indicates easier access); **(c)** *computational complexity*: resource requirements and runtime implications for training and inference (low signifies greater efficiency, while high requires more resources). When considering explainability, we describe the potential for given the necessity for human intervention. This standard is also seen when evaluating trustworthiness in model use. Moreover, while many applications are visually based, qualitative insights can vary by expert; what is interpretable for one may not be for another. Thus, we define such methods as expressing the potential for it to be explainable.

The trade-offs between these factors are evident across the methods. For example, the deterministic approaches in Chapter 2 are trained in such a way that they increase the potential for explaining the predictions through the properties of the model at a low computational cost, making them well-suited for smaller tasks where explainability is important. In contrast, more advanced techniques, such as the diffusion model in Chapter 5, can scale to large amounts of data, but with increased computational demand and limited capacity to understand how the model arrived at its output.

## 1.2 Contributions

This dissertation is based on research that has been presented and published at various conferences, workshops, and journals between 2022-2024. In summary, our primary contributions are as follows:

- In Chapter 2, we introduce a one-dimensional, adaptive wavelet layer whose parameters are learned through backpropagation. Additionally, we present a channel-separated attention scheme that dynami-

cally reweighs coefficients from the scattering transform, increasing the potential for explainability and improving performance on tasks with limited training data. This work was featured at the 10th International Conference on Learning Representations Workshop on AI for Earth and Space Science, the 36th Annual Conference on Neural Information Processing Systems Workshop on Tackling Climate Change with Machine Learning, and the 22nd Conference on Artificial Intelligence for Environmental Science at the 103rd Annual Meeting of the American Meteorological Society.

- In Chapter 3, we propose a transformer-based memory module that contextualizes the history of smaller, observed locations within spatial imagery, replacing the recurrence in traditional sequential attention methods. We show how these dynamic attention weights guide model predictions and can simplify model explanations. This classical components of this work appeared at the 36th Annual Conference on Neural Information Processing Systems Workshop on Gaze Meets Machine Learning and was published in the 226th Volume of Proceedings for Machine Learning Research.

- In Chapter 4, we design a transformer-based network for image-to-image translation and develop a method to quantify and compare the distribution of image sharpness across datasets and models. Additionally, we introduce a token attribution method for transformer-based models to trace the flow of information resulting from self-attention. This work appeared at the 41st International Conference on Machine Learning Workshop on Machine Learning for Earth System Modeling.

- In Chapter 5, we discover that single-step autoregressive diffusion models, trained on observational data without strong priors, can effectively learn long-range dynamics in variably chaotic environments. Additionally, we introduce the use of classical domain-specific diagnostics, such as Hovmöller and Wheeler–Kiladis diagrams, to provide novel insights into the modeling behavior of diffusion-based models, marking the first application of these analyses in this context. This work appeared at the 12th International Conference on Learning Representations Workshop on Tackling Climate Change with Machine Learning.

During this period, additional contributions were made to other publications, providing valuable insights that, while related, fall beyond the scope of this dissertation. These include a stochastic correlative learning algorithm to find optimal inputs through inverse optimization [27], a progressive cascade network that incrementally adds model complexity during training [31], a study on measuring the sharpness of AI

generated meteorological imagery (currently under review), and improving the vertical profiles of numerical models with uncertainty-based machine learning [32]. Collectively, these studies intersect with and support the dissertation's themes, providing additional perspectives and serve as complementary readings.

## 1.3 Reader's Guide

The chapters of this dissertation are divided by modeling techniques with non-overlapping applications in the Earth sciences. Each method is self-contained, with chapters presenting the central motivation, methodology, and supporting experimental analyses. Some chapters include vignettes to provide further intuition through additional, tangential experiments. These sections can be skimmed or revisited in more detail, but they are not central to the main application. A discussion of each method's impact and limitations is provided at the end of every chapter.

We begin with classical image processing by introducing ways to integrate wavelets into neural networks to improve feature extraction in Chapter 2. This chapter covers methods to learn parameterized wavelet functions and strategies for attending to their coefficients using soft spatial attention. The concept of attention carries into subsequent chapters. The first section in Chapter 3 offers essential background on attention, both as a biological process (from cognitive psychology and neuroscience) and as a computational model. Within this chapter, we introduce our neuro-inspired model of sequential attention. Evaluations are done on classical vision tasks, and demonstrated on a practical climate application.

Chapters 4 and 5 are slightly more domain-oriented, with methods designed to address research questions driven by the applications themselves. Transitioning from sequential attention to self-attention, Chapter 4 presents a transformer-based network for image-to-image translation. In this chapter, we also derive an explainability method for transformer models to guide domain experts in understanding model predictions. In Chapter 5, we shift focus to generative diffusion methods to study their long-range spatiotemporal dynamics. This chapter emphasizes the learning process itself, and offers a new perspective on modeling temporal dependencies beyond the scope of earlier attention-based models.

While there are data similarities in most chapters using satellite observations, the underlying task varies, with Chapters 2 and 4 focusing more on weather and Chapters 3 and 5 on climate. For the domain experts, Chapter 2 will cover gravity wave identification, estimation of the intensity of tropical cyclones, and short-range lightning prediction. In Chapter 4 we center on estimating composite radar reflectivity from satellite imagery across the United States. Shifting our focus toward climate, in Chapter 3 we identify indicators of

climate change from climate simulations of global temperature. Then in Chapter 5 we study the tropical waves modes that govern climate and subseasonal-to-seasonal predictability. Note that Chapters 2 and 3 were designed independently of extensive domain collaborations, making their applications more demonstrative, whereas Chapters 4 and 5 involved greater iteration with domain experts.

Finally, in Chapter 6, we discuss concluding remarks and insights gleaned. We also provide guidance that is relevant to the future of artificial intelligence and machine learning for Earth system science, including future research directions specific to this dissertation and perspectives on the field at large.

Immediately moving forward, recall that Table 1.1 offers a quick reference to identify which methods may be of most interest, based on whether they are deterministic or probabilistic, their primary applications, or their computational factors. Moreover, as each method is largely self-contained, we provide background information within each chapter rather than as a separate standalone chapter.

# Chapter 2

# Integrating Wavelets and Spatial Attention into Neural Networks

In machine learning, particularly in the context of processing high-dimensional data, the methods used for feature extraction and learning representations are crucial for task performance. Wavelets, stemming from signal processing, offer a powerful way to decompose functions into their frequency components, making them well-suited for capturing both local and global patterns in data. This chapter introduces wavelets and their integration into neural networks.

We begin in Section 2.1 by providing motivation for the use of wavelets in machine learning. We then introduce two methods that build on the foundation of wavelets. The first is a wavelet neural network (WaveNet, Section 2.2) that learns the frequency components of parameterized wavelet functions in an end-to-end manner. The second leverages the scattering transform (a cascade of fixed wavelet transforms) with a separation scheme to bring attention to independent wavelet coefficients and input channels (scattering network, Section 2.3).

With these methods, we demonstrate them on both simplified datasets and real-world atmospheric science applications, including the detection of atmospheric gravity waves, the estimation of tropical cyclone intensity, and the prediction of lightning occurrence from satellite imagery. Our results are compared to more traditional network architectures, each trained and optimized for task performance. Although the baseline models are not necessarily state-of-the-art, these comparisons highlight their limitations within our demonstrative applications and emphasize the benefits of our data-driven model development. Finally, in Section 2.4, we summarize the key takeaways, discuss limitations, and consider the broader impact of our findings.

Additional reading as it relates to this chapter can be found in the corresponding publications:

Stock, J., & Anderson, C. (2022). *Trainable Wavelet Neural Network for Non-Stationary Signals*. In ICLR 2022 Workshop on AI for Earth and Space Science, Apr, 2022.

Stock, J., & Anderson, C. (2022). *Attention-Based Scattering Network for Satellite Imagery*. In NeurIPS 2022 Workshop on Tackling Climate Change with Machine Learning, Dec, 2022.

## 2.1 Background and Motivation

Classical filters in signal processing are excellent tools to transform time-series data to the spatial and frequency domains, allowing us to localize spectral information. When paired with machine learning, this transformation is traditionally done during preprocessing using a pre-defined filter-bank. The result is then represented by magnitude spectrogram features and used as input to a neural network. Prior work from various applications [33, 18, 34] demonstrate this to be effective, but require extensive hyperparameter tuning on the number of frequency bins, duration, and overlap. Moreover, dominate frequencies within the underlying data may not be known a priori, which can yield inaccurate or uninterpretable results.

Convolutional neural networks (CNN)s provide a way to learn data-dependant filters and have shown to be effective by themselves for atmospheric science applications [35–37]. However, more recently, there has been work suggesting wavelet- and adaptive-based filters can yield greater performance. Examples include learning parameterized sinc functions to preform band-pass filtering [38], learning wavelet coefficients from a multiresolution lifting scheme [39], and with the scattering (or cascade of wavelet) transforms [40], a method that is of particular interest in Section 2.3.

The scattering transform has strong performance with relatively few training samples [41–44] due to its ability to build geometric invariants (e.g., to translations, rotations, and scaling) that are stable to the action of diffeomorphisms—a desirable trait due to the continuous change in atmospheric structure over time. This ultimately promotes sparse representation of data with a high degree of discriminability and can simplify downstream tasks [41, 45]. However, using a fixed filter-bank will yield an excess of coefficients that may not be relevant to every data sample. In such case, if we are not dynamically learning the wavelet parameters, we need to learn the importance of these individual coefficients.

The following sections introduce two methods of integrating wavelets into neural networks to address these limitations. Section 2.2 focuses on how we can alleviate exhaustive hyperparameter tuning by directly learning the parameters of a traditional wavelet function to model the underlying signal. The second method in Section 2.3 similarly leverages wavelets, but with fixed parameters, and instead learns how best to attend to individual coefficients in a data constrained manner.

## 2.2 Wavelet Neural Network

To learn a filter-bank specialized to fit non-stationary signals, we use a wavelet transform as the first layer of a neural network where the convolution is a parameterized function of the complex Morlet wavelet. We

will first introduce our approach in Section 2.2.1 and then provide an application to atmospheric gravity waves in Section 2.2.2, showing the network is quick to converge, generalizes well on noisy data, and outperforms standard network architectures.

### 2.2.1 Methodology

**Wavelet Transform** We define this as the use of local wavelike functions to present a signal in its time-frequency domain. Wavelets can be manipulated by moving to different locations on the signal or stretched and squeezed to cover different frequencies. The transform loosely quantifies the local matching of the wavelet and the underlying signal, whereby a large transform value is observed if the location and scale of the wavelet match the signal. The integral wavelet transform is performed with an inner product of a function $x(t)$ and mother wavelet $\psi(t)$, at a scale $s \in \mathbb{R}^+$ and translational value $\tau \in \mathbb{R}$, as

$$W_x(s,\tau) = \frac{1}{|s|^{1/2}} \int_{-\infty}^{\infty} x(t)\, \psi^*_{f,w}\Big(\frac{t-\tau}{s}\Big)\, \mathrm{d}t. \tag{2.1}$$

For this work we use the complex Morlet wavelet with frequency $f$ and width $w$, given (with labeled components) by

$$\psi_{f,w}(t) = \underbrace{\big(s_t(2\pi)^{-\frac{1}{2}}\big)^{-\frac{1}{2}}}_{\substack{\text{Normalization}}} \underbrace{\exp\big(\mathrm{i}2\pi f t\big)}_{\substack{\text{Complex}\\ \text{sinusoid}}} \underbrace{\exp\Big(-\frac{t^2}{2s_t^2}\Big)}_{\substack{\text{Gaussian}\\ \text{envelope}}}, \tag{2.2}$$

where $s_t = (2\pi s_f)^{-1}$ with $s_f = fw^{-1}$, denoting the standard deviation (resolution) of the wavelet transform in the temporal and in the spectral domains, respectively. The choice of the wave function is determined by the structural similarities with the data. Figure 2.1 illustrates this with the sinusoidal components separated in the real and imaginary domain.

In Figure 2.2, we show the continuous wavelet transform applied to a frequency-swept cosine signal, or chirp, with linearly increasing frequencies ranging from 1 to 10 Hz. A total of 64 wavelets are initialized logarithmically with frequencies in the range $[1, 15)$ Hz. The log-scale power (square of the magnitude) scalogram reveals when these frequencies occur in the signal over time.

**Network Architecture** WaveNet uses a wavelet transform *as the first layer* of a neural network where the convolution is a parameterized function of a mother wavelet. Specifically, we use $\psi_{f,w}(t)$ defined previously where $\mathbf{f}$ and $\mathbf{w}$ are vectors of trainable parameters learned through backpropagation. The vector notation

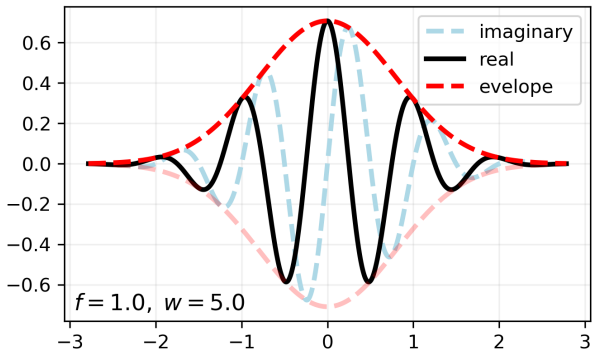



**Figure 2.1:** Individual components of the complex Morlet wavelet with frequency and width.

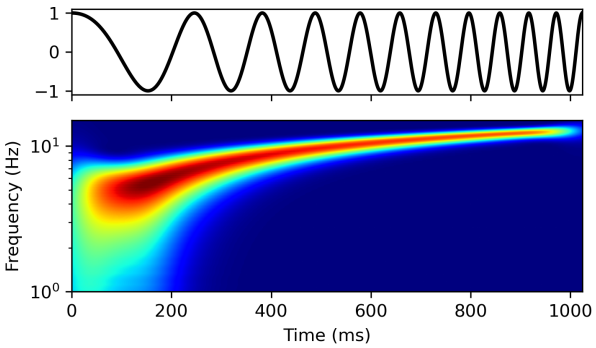



**Figure 2.2:** Chirp signal (top) and power frequency (bottom) continuous wavelet transform.

allows us to have multiple wavelet filters. In the forward pass, the real and imaginary convolutional output are combined and the magnitude propagates to subsequent layers. Input signals are not standardized, thus, batch normalization is applied following the transform layer to regularize the data and stabilize training. Thereafter, the values are input to zero or more fully-connected layers with tanh nonlinearities before the final linear output layer.

Using too small a learning rate, $\eta$, during training will yield very small updates to the wavelet parameters. As such, different learning rates are set for $\mathbf{f}$ and $\mathbf{w}$ in the transform layer and all other parameters, $\theta$, in subsequent layers. Specifically,

$$(\mathbf{f}, \mathbf{w})^{(k+1)} = (\mathbf{f}, \mathbf{w})^{(k)} - \eta_0 \nabla_{(\mathbf{f},\mathbf{w})} \mathcal{L}(\mathbf{f}, \mathbf{w}, \theta), \tag{2.3}$$

and for the rest of the parameters,

$$\theta^{(k+1)} = \theta^{(k)} - \eta_1 \nabla_{\theta} \mathcal{L}(\mathbf{f}, \mathbf{w}, \theta), \tag{2.4}$$

where the learning rates are related by $\eta_0 = \eta_1 \cdot 1e3$. This change emphasizes the wavelet parameters and significantly stabilizes training and improves performance. However, too large or an incorrect weight update to Equation (2.3) can lead to negative or exceedingly large values, which is undesirable for the wavelet transform. Therefore, after each update step, we clip the values of $\mathbf{f}$ and $\mathbf{w}$ between $[0.5, 30]$ and $[4, 15]$, respectively. We find these value, although easily adaptable to a particular problem, to work best in practice and be more stable than gradient clipping.

**> vignette (1):** ***simplified example on synthetic data***

To demonstrate the efficacy of WaveNet, we train it to classify synthetic data composed of a background signal and time independent events of a predefined frequency. The intended result is to have the transform layer fit the event frequencies such that a single linear output layer can classify the samples. Data are generated for two classes using a sampling rate of 256 Hz:

$$y_{A,B}(t) = \sin(2\pi f_b t + \varphi_0) + \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(t-\mu)^2}{2\sigma^2}\right) \sin(2\pi f_e t + \varphi_1) + \varepsilon(t). \tag{2.5}$$

The background signal is a sine wave with $f_b = 9$ Hz and added normally distributed noise $\varepsilon \sim \mathcal{N}\left(-0.5, 0.5^2\right)$. Each class differs in its event frequency, $f_e$. Specifically, class A has $f_0 = 5$ Hz and class B with $f_1 = 15$ Hz. These are localized in time via a Gaussian envelope that is randomly shifted and centered by $\mu$ with $\sigma = 0.8$. All data samples have a duration of $0.80\,\mathrm{s}$ (i.e., 205 time steps) and are randomly phase shifted by $\varphi_0$ and $\varphi_1$ between $[0, 2\pi]$. Once generated, data are partitioned into training and test splits with equal class distributions, totaling 240 and 60 data samples, respectively.

A simple WaveNet is initialized with two wavelet filters having frequency values of 8 and 12 Hz and a linear output layer to discriminate between the two classes. A constant non-trainable width value of $w = 10$ is set and used for both filters. The network optimizes the cross entropy loss on the training data using adaptive moment estimation (Adam). Within the first 60 epochs, the network starts converging to a higher classification accuracy; however, we continue training for a total of 700 epochs to visualize how the wavelet parameters and network performance changes over time.

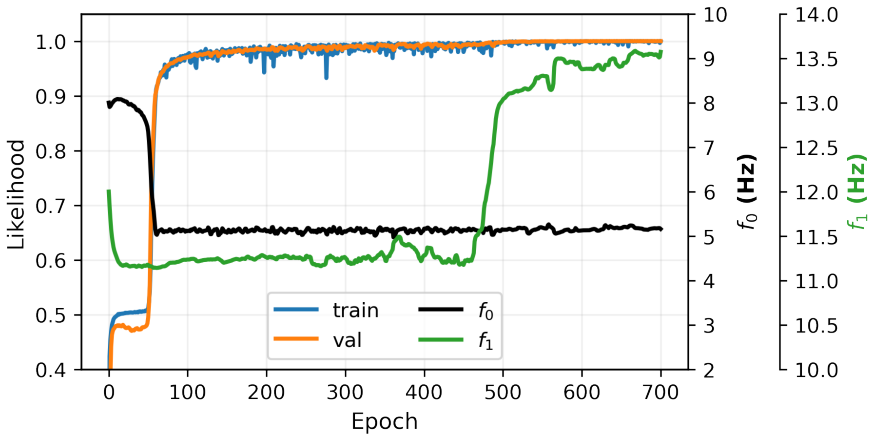



**Figure 2.3:** Training and validation curves and central frequency values of WaveNet during training on simplified data (test accuracy of 100%).

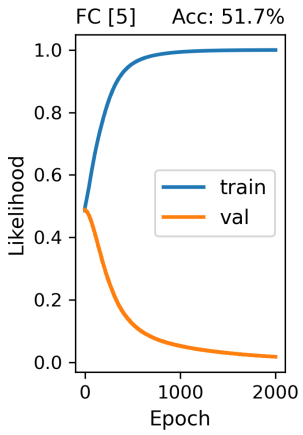



**Figure 2.4:** Fully-connected network with test accuracy.

Figure 2.3 illustrates the likelihood during training as well as the change to the adaptive frequencies, $f_0$ and $f_1$. At 60 epochs $f_0$ rapidly approaches a value of 5 Hz, matching the target event frequency for class A. As a result, the likelihood simultaneously rises to a value near one for both training and validation, yielding $98\%$ classification accuracy. With only two classes, the network performs well when matching a single event frequency, but the training loss continues to oscillate. After 480 epochs we observe the training loss smooth out and the other wavelet frequency, $f_1$, approach the desired value of 15 Hz and the network achieves $100\%$ test accuracy.

The observation of $f_1$ locking onto the target frequency is very intriguing, considering the validation and training loss only marginally change. This is a binary classification problem, so it makes sense to only need to identify a single frequency for accuracy predictions. However, we speculate the effect of $f_1$ converging is actually a special form of *grokking*: a phenomenon observed in neural networks, where after an initial phase of overfitting (or memorization), the model suddenly achieves perfect generalization [46, 47]. In our case, we are not overfitting, but the single frequency parameter groks to its target value after a large number of epochs, ultimately stabilizing the loss. This particular observation is not well understood in machine learning, and warrants future study.

We conduct a second experiment with a standard fully-connected network of one hidden layer of 5 units and nonlinear activations. Note, a non-exhaustive architecture search was performed. We use the same training and test partitions, and find the network to quickly overfit on the training data (Figure 2.4). This occurs specifically when noise, $\varepsilon$, is added to $y_{A,B}(t)$, and the network severely lacks generalization and only achieves $\sim 50\%$ accuracy on the test data. Thus, making WaveNet more desirable, in terms of generalizability and learning performance, for data with quasiperiodic variations in time.

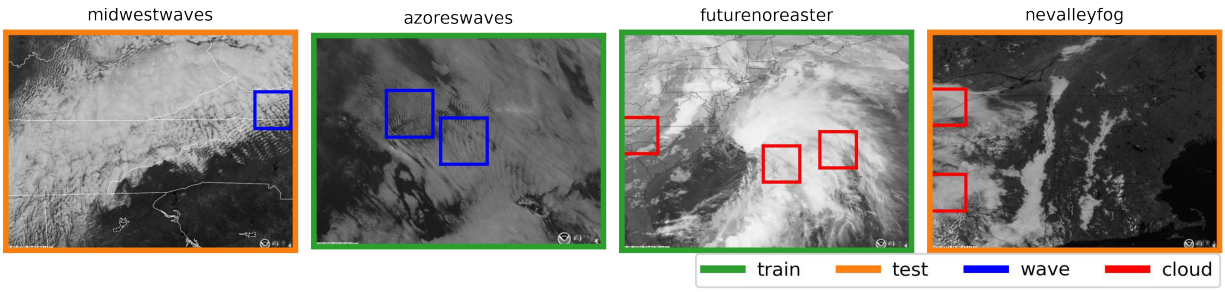



**Figure 2.5:** A subset of satellite imagery frames with labeled gravity wave and cloudy scenes. Individual boxes in red and in blue represent the patch of the image used for training (in green) and test (in orange).

### 2.2.2 Identifying Gravity Waves

Automatic detection of atmospheric gravity waves from satellite imagery is of interest to assimilate and improve the accuracy of numerical weather prediction models [48]. Gravity waves are initiated by disturbances to the density structure of the atmosphere and restored via gravity and buoyant forces. An intuitive analogy would be similar to a rock being thrown in a pond, causing ripples to spread outward. In the case of gravity waves, the rock is a column of displaced air, and the pond is the atmosphere itself, with the waves propagating through it—often visible in the clouds themselves.

#### Dataset Details

Gravity wave and cloud data are acquired from the Cooperative Institute for Research in the Atmosphere (CIRA)'s Geostationary Operational Environmental Satellite (GOES)-16/17 Loop of the Day[1]. Each loop includes monochromatic imagery from animated GIFs, comprising seven gravity wave and seven cloudy animations, with standard pixel-value intensities between $[0, 255]$ (i.e., not true reflectance/brightness temperatures) standardized to have a max value of one. However, all animations are captured using the visible $0.64\,\mu\mathrm{m}$ wavelength band from the Advanced Baseline Imager (ABI) [49]. The GOES-R series satellites have a nominal resolution of $2\,\mathrm{km}$, but the animations are zoomed and cropped at different aspects with no specific map projection.

For all $14$ animations, we hand select and label patches of size $128 \times 128$. Not all images contain gravity waves or clouds, so labeling known patches helps create a more accurate, though not fully representative, dataset. Figure 2.5 shows frames from a subset of animations used for training and testing with corresponding labels. We shuffle and truncate each dataset to have an equal number of per class samples with separate animations held out for testing.

Individual patches are preprocessed so that intensity values across the spatial domain become candidate 1D samples. This differs from the typical time-series data used in time-frequency analysis with wavelet transforms. However, we can treat these 1D intensity signals as time-series data with a predefined sampling rate. Essentially, vertical and horizontal pixel-intensity slices are extracted from each patch at all $128$-$x$ and $128$-$y$ coordinates, and each slice is assigned the patch label. After slicing, we are left with training data of size $165{,}376 \times 1 \times 128$ and testing data of size $33{,}792 \times 1 \times 128$.

[1] https://rammb.cira.colostate.edu/ramsdis/online/loop_of_the_day/

Figure 2.6 is an example gravity wave patch, showing a subset of five vertical and horizontal intensity levels at the red and blue dashed lines. The many other slices are omitted for visual simplicity. We can see clear periodicity with frequencies that we hope to learn with WaveNet.

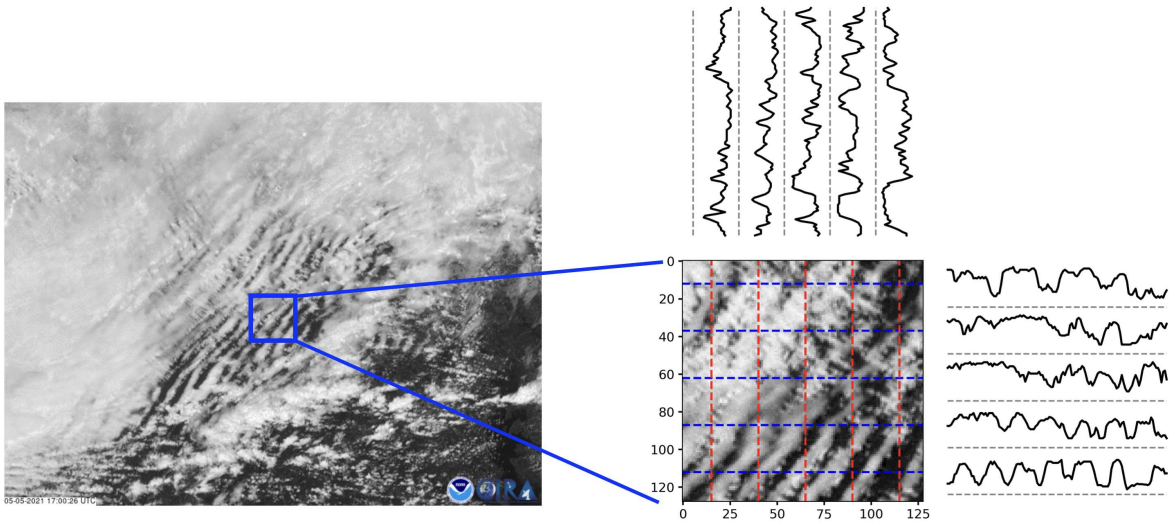



**Figure 2.6:** Sample gravity wave broken into vertical and horizontal slices, representing a subset by the red and blue dotted lines, respectively. The corresponding signals for these subsets are shown on the top and to the right.

### Training Details

We initialize WaveNet with 20 wavelet filters composed of linearly spaced frequencies between $[1.5, 25]$ Hz and a single hidden layer of five nonlinear fully-connected units. A width parameter of $w = 10$ is held constant and not learned during training. Through a hyperparameter grid search, we find the network need only be trained for 10 epochs (bs $= 128$) using $\eta_0 = 0.1$ and $\eta_1 = 0.0001$.

We also train and evaluate other fairly standard neural network architectures, namely: a fully-connected network (fc-net), 1D convolutional network (conv-net), and filter-bank network (bank-net). As with WaveNet, the training parameters for each network are found through a preliminary hyperparameter search. We find that using two hidden layers with 20 units each performs best for the fc-net. The conv-net is found to have two convolutional layers with the first layer having four filters and the second with eight, both with a kernel size of three, and separated by max pooling. Lastly, the bank-net is most similar to WaveNet in that we use a fully-connected network with one hidden layer of five units, but with preprocessed wavelet transformed

samples from a filter-bank of $20$ wavelets that have linearly spaced frequencies between $[1.5, 25]$ Hz, i.e., the wavelet parameters are static.

Each network is trained and evaluated on the test data $10$ times with different random weight initializations. This is to capture a more robust view of network performance and to alleviate the potential of settling in a local minima. Accuracy is the primary metric of interest due to having a binary classification task with an equal number samples belonging to each class. However, we do also report on precision and recall from the test data to bring insight to the predictions of both classes.

**Main Findings**

Figure 2.7 presents the aggregated test accuracies across all training trials for each network with the gravity wave dataset. The fc-net exhibits similar behavior to the simplified example in our vignette. Specifically, the network is prone to overfit on the training data after the first five epochs and performs the worst overall. This is likely due to the intrinsic noise and pixel-level variability in the data. If we consider the spatial context, and train the conv-net on the data, then we observe better generalization and an increase in accuracy of $12\%$ over fc-net. Adding additional fully-connected layers or increasing the depth of convolutions with the intent of identifying fine-grained patterns in the conv-net results in overfitting.

WaveNet and bank-net also exploit the spatial patterns, but rather through *time*-frequency analysis with the wavelet transform, where time is actually in the spatial domain. The bank-net reaches a slightly higher accuracy, with a value of $80.57 \pm 1.12\%$, over the conv-net, but the improvement is not significant. The proposed WaveNet achieves the highest accuracy of $81.74 \pm 3.03\%$ and a maximum test accuracy of $88.32\%$. It is important to note that the WaveNet exhibits a large variability in performance with changes to the initial weights. This is as a result of training instability with the network occasionally showing relatively large oscillating loss values and the lack of generalization to unseen data (not shown within). However, the top performing network is of interest and studied in more detail from hereon.

Of the $16{,}896$ per class test samples, those labeled as a cloud have a higher recall value of $0.96$, whereas the gravity wave samples have a recall of $0.80$, indicating the occurrence of more false negatives. While this is seemingly undesirable, it can be reasoned as not every slice within a patch intersects with a gravity wave—especially as animations evolve over time. We validate this claim by viewing the network's predictions over the entire patch. Recall that the network predicts on individual vertical and horizontal slices of an image patch. Therefore, to have a more comprehensive view of the prediction, we run a forward pass of every

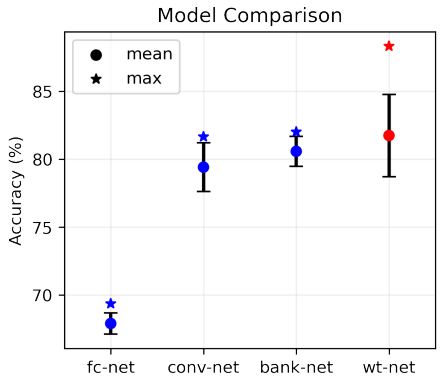



**Figure 2.7:** Test statistics of models trained on gravity wave data.

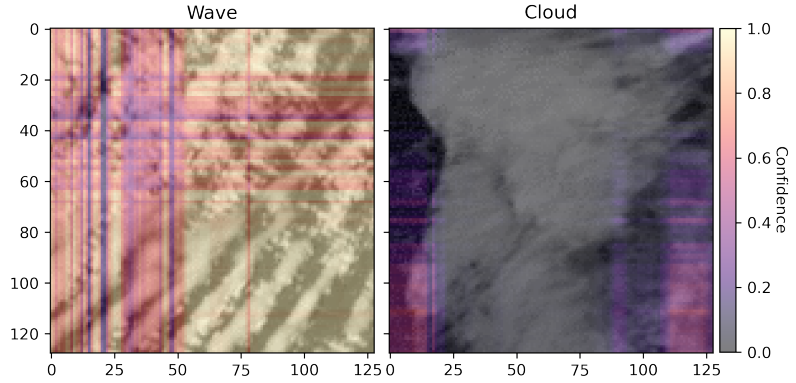



**Figure 2.8:** Confidence map on test wave and cloud patches. Higher values indicate greater confidence of a wave.

sample in a patch and compute a confidence map. Evaluating $p(y_i = 1_{wave} \mid x_i; \mathbf{f}, \mathbf{w}, \theta)$ from the softmax (logistic) output of each sample, we calculate the outer product with the vector of all row and column values to produce a single matrix outlining the network confidence across the entire patch. A pixel-value toward one indicates a greater probability of being labeled as a gravity wave.

Figure 2.8 overlays this confidence map on a single test patch of gravity wave and cloud images. In the lower-right quadrant of the gravity wave map (left), we see high confidence in the network's prediction of gravity waves. The least confident region is in the upper-left quadrant, which has high cloud coverage and no visible periodicity. The cloud image (right) has a max confidence of $0.62$ around the boarders where there are slices with both cloud and land mass visible. However, there are no wave-like slices that yield a high confidence value.

As for the transform layer within WaveNet, there are multiple filters that converge on nearly the same frequency values, i.e., $2.4$, $13.2$, and $29.5$ Hz. The majority of low-frequency values (between $1.5$-$7.0$ Hz) do not move far from their initial value, which we speculate to be representative in the original dataset. Reducing the number of filters could enable the network to learn more unique frequency values, but we find the additional filters to improve performance.

## 2.3 Scattering Neural Network

To build an architecture that more closely aligns with the visual interpretations of satellite imagery done by forecasters, we incorporate attention into the early layers preceding the scattering transform. Attention mechanisms, inspired by the human visual system, help identify salient regions in complex scenes [51].

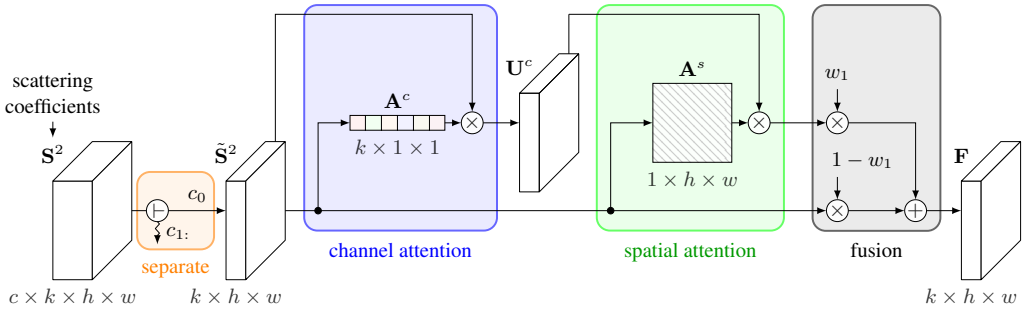



**Figure 2.9:** Network architecture illustrating the separation of attention modules on the scattering transform. The left most block represents the output of the scattering transform on the input. The separate operator isolates a single channel and passes the normalized scattering coefficients, $\tilde{\mathbf{S}}^2$, through channel attention and spatial attention before fusion. There are $C$ total attention modules in the network. Figure modified from [50].

Recent computer vision studies have shown attention to increase performance and interpretability while also improving confidence of post hoc explainability methods [52, 50]. For a more in-depth background of attention, see Chapter 3. The studies most similar to this work are those by [53, 44]. In [53], residual layers mix the input channels before applying attention and [44] applies a scattering attention module after each step in a U-Net. Our approach outlined in Section 2.3.1 differs in that we introduce a separation scheme that applies attention to individual input channels that directly follow the scattering transform.

We demonstrate our method and show promising results for estimating tropical cyclone intensity (Section 2.3.2) and predicting the occurrence of lightning (Section 2.3.3) from satellite imagery. Our results are followed by an explanation of what the network has learned in Section 2.3.4.

### 2.3.1 Methodology

Figure 2.9 illustrates the primary components of our network, starting with our output of the scattering transform and showing an attention module separated by input channel. The implementation and design choice for each part is described in detail below.

**Scattering Transform** Scattering representations yield invariant, stable (to noise and deformations), and informative signal descriptors with cascading wavelet decomposition using a nonlinear modulus followed by spatial averaging. Using the Kymatio package [54], we compute a 2D transform with a predetermined filter-bank of Morlet wavelets at $J = 3$ scales and $L = 6$ orientations. This is similar to the wavelets defined in Section 2.2.1, but now in two dimensions with orientations. For each input channel, we apply a

second-order transform to obtain the scattering coefficients $\mathbf{S}^2$. These channels are processed independently and combined later in the network.

The specifics of the scattering transformed are defined as follows. Let the wavelet transform of a 2D signal, $x(u)$ with $u$ denoting the spatial index, at scale J be defined as

$$\mathcal{W}_J x(u) \coloneqq \left\{x * \phi_{2^J}(u)\ ,\ x * \psi_\lambda(u)\right\}_{\lambda \in \Lambda_J}, \tag{2.6}$$

where $\psi_\lambda(u) = 2^{-2j}\psi(2^{-j}r^{-1}u)$ with $\lambda = 2^j r$ for $0 \leq j < J$ and $r \in G^+$ as the discrete, finite rotation group of $\mathbb{R}^2$ with $L$ equally spaced angles from $[0, \pi)$. The traditional mother wavelet, $\psi(u)$, in the scattering transform is the Morlet wavelet with a scaled Gaussian lowpass filter, $\phi_{2^J}(u) = 2^{-2J}\phi(2^{-J}u)$. A wavelet transform is translation covariant, and thus invariance measures are extracted by computing a nonlinear complex modulus, $|x + iy| = \sqrt{x^2 + y^2}$, and averaging the result. The information lost during averaging is restored by applying a new wavelet decomposition with scales $j_1 < j_2$, producing new invariants.

By following an iterative scheme we can compute $m$-th order coefficients, although here we compute up to the second order as higher orders have negligible energy [41]. The zeroth-order coefficient is computed as $S^0 x(u) = x * \phi_{2^J}(u)$ and downsampled by a factor of $2^J$. To recover the high-frequency information, we perform our first wavelet transform, apply a nonlinear modulus, and average again. Formally, the first-order coefficients are found by

$$S^1 x(\lambda_1, u) \coloneqq |x * \psi_{\lambda_1}| * \phi_{2^J}(u). \tag{2.7}$$

The resulting feature maps have the same resolution as $S^0$ but with $JL$ channels. Second-order coefficients are computed similarly on $S^1$ using all rotations but for smaller coefficients, denoted by

$$S^2 x(\lambda_1, \lambda_2, u) \coloneqq ||x * \psi_{\lambda_1}| * \psi_{\lambda_2}| * \phi_{2^J}(u), \tag{2.8}$$

which results in feature maps with $\frac{1}{2}J(J-1)L^2$ output channels. Thus, if we assume $x$ to be a tensor of size $(B, C, W, H)$, then the output via a second-order scattering transform, $\mathbf{S}^2$, with scale $J$ and $L$ angles will have size $(B, C, 1 + LJ + \frac{1}{2}J(J-1)L^2, W/2^J, H/2^J)$. Note, for brevity, in the following sections we use $W \times H$ to denote the spatial dimension that actually occur over $W/2^J \times H/2^J$.

**Channel Separation** Local attention methods routinely process their input using all the channel information at once, e.g., feature maps from RGB color channels. However, the result of the scattering transform yields a 5-dimensional tensor, $\mathbf{S}^2$, where each channel, $C$, in the input has their own set of $K$ scattering coefficients. Rather than stacking the result and passing them all through the subsequent layers together, we propose to first separate the input channels and process the coefficients individually. This creates $C$ new attention modules, each with independent weights, that are processed in parallel. By following this separation scheme we add the benefit of localizing patterns in the input before joining high-level features. Thus, the interpretation of attention over individual input channels is improved significantly, especially if the channels have different meaning, e.g., in satellite imagery this can be visible or infrared channels or some other derived, physical product.

**Channel Attention** Channel attention is used to inform the spatial attention module before fusion via feature recalibration. Specifically, the network learns to use the spatial information over the $K$ channels to selectively highlight the more informative coefficients from the less useful ones. Not only does this offer a performance improvement to our network, but it also adds an additional layer of interpretability with channels corresponding to particular coefficients.

Following the design of the squeeze and excitation block in [55], this attention module emphasizes the local channel information of individual inputs. We use $\tilde{\mathbf{S}}^2 = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_K]$ as the normalized coefficients from the separated input channel of $\mathbf{S}^2$. First, we *squeeze* $\mathbf{s}_k$ to obtain a global information embedding $\mathbf{z} \in \mathbb{R}^K$ via global average pooling, where the $k$-th element, $z_k$, is

$$z_k \coloneqq \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} s_k(i,j). \tag{2.9}$$

Thereafter, we aggregate the embedding with the *excitation* operation to adaptively recalibrate channel-wise features. This results in our *channel attention weights* $\mathbf{A}^c \in \mathbb{R}^K$ with scalar elements, $a_k$, computed as

$$\mathbf{A}^c \coloneqq \sigma(g(\mathbf{z}; \mathbf{v}, \mathbf{w})) = \sigma(\delta(\mathbf{z}\mathbf{v})\mathbf{w}), \tag{2.10}$$

where $\sigma$ and $\delta$ refer to the sigmoid and ReLU functions, $\mathbf{v} \in \mathbb{R}^{K \times \frac{K}{r}}$, and $\mathbf{w} \in \mathbb{R}^{\frac{K}{r} \times K}$. Weight matrices $\mathbf{v}$ and $\mathbf{w}$ are initialized without bias parameters and use a reduction ratio of $r = 16$. This ratio value was shown by [55] to be a sufficient starting parameter across multiple experiments. The final output of the channel

attention weighs the normalized coefficients with the scalar elements $a_k$ to get $\mathbf{U}^c \in \mathbb{R}^{K\times H\times W}$, where the $k$-th filter is given by $\mathbf{u}_k \coloneqq a_k \mathbf{s}_k$.

**Spatial Attention** We use spatial attention to highlight the salient features in the spatial resolution of independent input channels. This differs from most computer vision problems with RGB imagery that only have one heat map for the full image. Instead, our network provides a more transparent interpretation of how the spatial information in each input channel is used to form a prediction.

We implement a spatial attention module that predicts the importance of regions within the scattering coefficients based on the image context, similar to [56, 50]. We first apply a pointwise convolution to our normalized coefficients with $r \times 1 \times 1$ filters, using a reduction ratio of $r = 16$, to compress channel dimensionality. Spatial context is then extracted from the resulting feature map by convolving three dilated convolutions of size $3 \times 3$ with dilation factors of one, two, and three. These dilations increase the receptive field while preserving the input resolution. This allows us to stack the four feature maps to create a $4r \times W \times H$ tensor. This is reduced via a pointwise convolution using a $1 \times 1 \times 1$ filter (per channel dimension) to yield the final *spatial feature map* $\mathbf{A}^s \in \mathbb{R}^{1\times W\times H}$.

The spatial feature map and channel weighted coefficients are multiplied together to get a complete local attention map, $\mathbf{U}^s \in \mathbb{R}^{K\times W\times H}$, as given by

$$\mathbf{U}^s \coloneqq \mathbf{A}^s \odot \mathbf{U}^c, \tag{2.11}$$

where $\odot$ is the Hadamard product. At a high level, $\mathbf{U}^s$ contains all the normalized scattering coefficients where each coefficient (i.e., $k$-th channel) is weighted and spatially scaled to a localized region. This information is used downstream to highlight the positive or negative features of the scattering transform.

**Feature Fusion** The normalized scattering coefficients and local attention maps are combined in the fusion block at the end of an attention module. We follow a method similar to [57] and [50] to get an output from the weighted average of features. A single trainable parameter, $w_1$, with an initial value of $0.5$ is trained to assign contributions of each pathway. Mathematically,

$$\mathbf{F} \coloneqq w_1 \mathbf{U}^s + (1 - w_1)\tilde{\mathbf{S}}^2, \tag{2.12}$$

where $w_1$ is clamped on the interval $[0, 1]$ after each update to ensure the contributions sum to one.

The result of applying attention to the scattering coefficients of each input channel yields $C$ output filters, $\mathbf{F}$, that are stacked to $\mathbf{U}^f \in \mathbb{R}^{C\times K\times W\times H}$. Following this could be any task specific transformation, e.g., additional convolutions, upsampling, residual connections, etc., but for our tasks we show how to design a regression and classification head to have relatively few trainable parameters. Specifically, we reshape $\mathbf{U}^f$ to have $CK$ channels, which we reduce to $16$ via a pointwise convolution. This effectively combines the high-level features of each input channel. The feature maps are flattened and input to a layer with $8$ fully-connected units before a single linear output. After the convolutional and fully-connected layers is a ReLU activation for added nonlinearity.

**Table 2.1:** Experimental results using $n$ training samples and $p$ parameters.

| $n \downarrow\ p \rightarrow$ | Scattering (51.8K) | ResNet18 (11.2M) | MobileNetV3 (1.5M) | Conv. (268.2K) |
|---|---|---|---|---|
| TC Intensity, rmse $(R^2)$ | | | | |
| 1000 | **15.83 (0.59)** | 16.47 (0.56) | 56.85 (-4.28) | 17.51 (0.50) |
| 5000 | **12.01 (0.76)** | 14.30 (0.67) | 55.18 (-3.97) | 13.34 (0.71) |
| 10000 | **10.98 (0.80)** | 11.85 (0.77) | 21.13 (0.27) | 13.81 (0.69) |
| 30000 | **10.35 (0.83)** | 10.74 (0.81) | 13.07 (0.72) | 11.68 (0.78) |
| 47904 | **9.33 (0.86)** | 10.55 (0.82) | 11.90 (0.77) | 11.67 (0.78) |
| Lightning Occurrence, acc. (F1) | | | | |
| 1000 | **86.04 (0.85)** | 73.68 (0.74) | 62.46 (0.39) | 78.27 (0.74) |
| 5000 | **88.01 (0.87)** | 87.59 (0.87) | 68.82 (0.55) | 82.35 (0.82) |
| 10000 | **88.87 (0.88)** | 86.33 (0.85) | 81.46 (0.83) | 84.37 (0.84) |
| 50000 | **89.58 (0.89)** | 89.20 (0.88) | 87.49 (0.87) | 87.99 (0.87) |
| 212604 | 90.46 (0.90) | **90.51 (0.90)** | 86.87 (0.88) | 89.57 (0.89) |

### 2.3.2 Estimating Tropical Cyclone Intensity

Tropical cyclones are among the most devastating natural disasters, causing billions of dollars of damage and significant loss of life every year. Predicting the track or path of these cyclones is a well studied topic, but there is still an imperative need to improve upon the forecast of intensity [58]. The NASA Tropical Storm Wind Speed Competition [59] was released to study new automated and reliable methods of forecasting intensity. The data are single-band infrared images (i.e., band-13 or $10.3\,\mu\mathrm{m}$) captured by the Geostationary Operational Environmental Satellite (GOES)-16 Advanced Baseline Imager (ABI) [49], with pixel values

representing heat energy in the infrared spectrum, normalized to grayscale. We leverage the temporal relationships of previous timesteps up to the point of prediction to estimate the maximum sustained surface wind speed.

The topical cyclone dataset contains a collection of satellite imagery for over 600 storms in the Eastern Pacific and Atlantic Oceans from years 2000 to 2019. Test data consists of imagery from storms not included in the training data as well as held out samples from later in a storm's life cycle. Furthermore, as observations from temporal data are not independent, we aim to reduce the implications of autocorrelations (i.e., from trends and seasonality) by leaving out imagery from the last $20\%$ of each storm to create the validation set.

To extend from the single channel imagery to multi-channel inputs we leverage the temporal relationships of previous timesteps up to the point of prediction. Three frames separated by a nine step interval (i.e., $t-18, t-9$, and $t$) are stacked to create a $3 \times 128 \times 128$ input sample using the last frame's wind speed (intensity) as the target value. An example is shown in Figure 2.10. Inputs are created following the next $t+1$ timestep and repeated over all datasets yielding $47{,}904$ training, $7{,}119$ validation, and $37{,}913$ test samples.

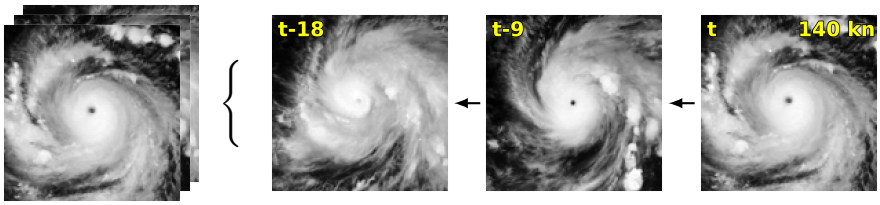



**Figure 2.10:** Input data of size $(1 \times 3 \times 128 \times 128)$ with stacked channel-wise timesteps (annotated in yellow). The target variable corresponds to the intensity of the last frame, i.e., at time $t$.

We initialize our scattering network to predict the target intensity from these channels. Individual samples are min-max normalized to the interval [0,1] to stabilize the result of the scattering transform. Target wind speeds are z-score normalized to have zero mean and unit variance using the statistics of the training data. Predicted and target values are unnormalized after inference for evaluation.

When subsampling the training data to create smaller datasets, i.e., values of $n$, we define $m = 11$ equally spaced boundaries, $B = \{r : r = 15 + 17k \mid k = m-1, m-2, \ldots, 0\}$, such that $n$ total samples that comprise the entire training dataset are loosely divided into groups following $B_i \leq t > B_{i+1}$ for targets $t$. Data within each group are randomly sampled and may borrow from the $B_{i+1}$ boundary to ensure an equal distribution of target wind speeds. Table 2.1 defines the values of $n$ used in our experiments, and range between $n = 1000$ to $n = 47{,}904$. The test and validation data are not resampled for evaluations.

**Main Findings**

The state-of-the-art achieves a root-mean-squared error (RMSE) of $6.26\,\mathrm{kn}$ (or knot) with an ensemble of 51 models, including convolutional and recurrent networks as well as vision transformers, each trained with different augmentations and validation schemes [60]. We omit a direct comparison as interpreting these models would be increasingly difficult due to the complexity introduced by model diversity and objectives. As such, we compare our proposed network, with significantly fewer parameters, with other, more traditional network architectures.

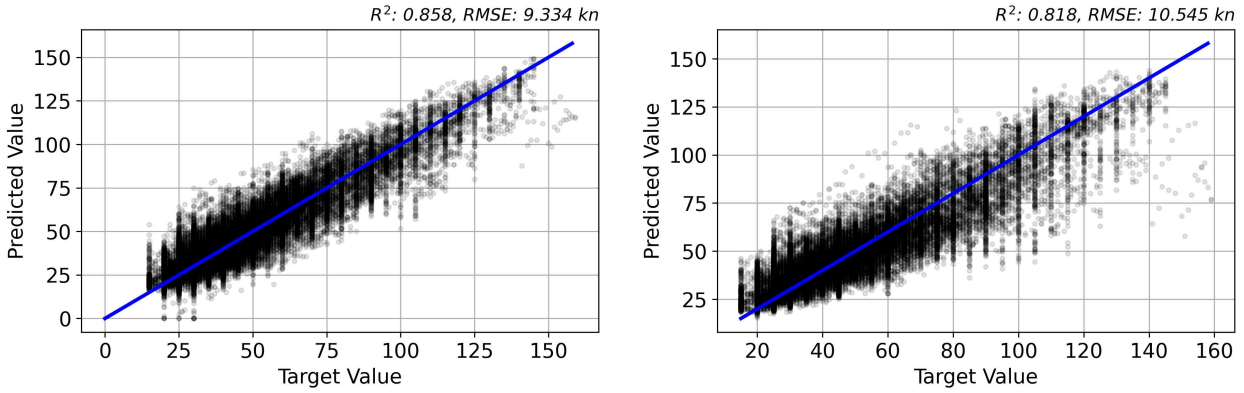
Our scattering network performs best overall with a minimum RMSE of $9.33\,\mathrm{kn}$ when using all available data for training. This is $12.97\%$ lower than the closest competitor, ResNet18, and $27.49\%$ and $25.03\%$ lower than MobileNetV3 and a standard convolutional network, respectively. A summary of performance for each model with its parameter and data size is reported in Table 2.1. Overall, the competing networks are more prone to overfit or lack the complexity to generalize, especially as the training size, $n$, decreases and for high-wind events. By leveraging the high-level features from the scattering coefficients, we maintain competitive performance even with $n = 5000$ training samples.

In Figure 2.11, we show a regression summary from our best network compared to ResNet18 when training on all available data. A perfect fit would follow the linear, blue line. The greatest errors are observed with the highest intensity samples for both our network and ResNet18. Target wind speeds $>140\,\mathrm{kn}$ have an RMSE $= 51.630\,\mathrm{kn}$ from ResNet18 as compared to an RMSE $= 27.231\,\mathrm{kn}$ from the scattering network. Interestingly, ResNet18 also strongly overestimates lower ($<40\,\mathrm{kn}$) storm events, whereas the scattering network has a lower spread and variance overall.

**Network Interpretations**

Local attention features, both spatial $\mathbf{A}^s$ and channel $\mathbf{A}^c$, can be visualized for each input channel, providing additional insights into the network's predictions. In Figure 2.12, we display a particular example (the same sample shown in Figure 2.10), analyzing input channels $t-18$, $t-9$, and $t$. For each channel, we display feature sensitivity through integrated gradients (IG), spatial attention superimposed on the input, and channel attention weights corresponding to the first- and second-order coefficients. Further details on the visual computations can be found in Section 2.3.4.

In this example, spatial attention $\mathbf{A}^s$ highlights particular patterns of the storm's structure. Specifically, with higher weights (in red) concentrated near the storm's eyewall, where the strongest winds typically occur.




**(a)** Scattering Network

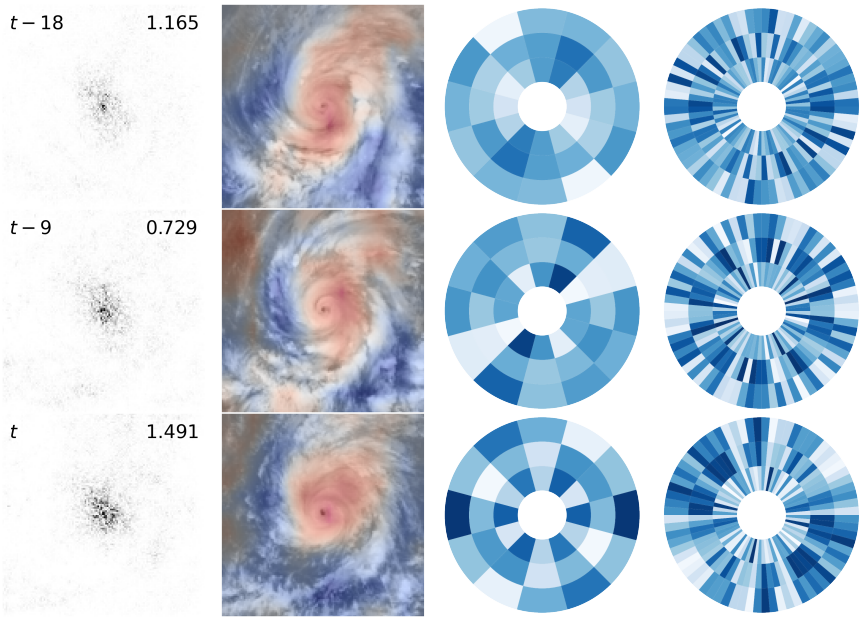
**(b)** ResNet18

**Figure 2.11:** Regression summary, highlighting the target vs. predicted wind speeds from the **(a)** proposed scattering network and **(b)** ResNet18, trained using all available data to estimate tropical cyclone intensity.

Notably, the highest weights are observed along the inner rainband at $t-18$ and $t-9$, and around the eyewall at $t$. Lower attention values (in blue) are generally found in the regions between rainbands, reflecting the network's focus toward the primary features of the cyclone.

Channel attention $\mathbf{A}^c$, shown in columns (c) and (d) of Figure 2.12, offers additional insights into the significance of scattering coefficients across different scales and rotations of the wavelets. The angles are symmetrically distributed around $[0, \pi)$ on the unit circle, with smaller polar radii representing larger scales (capturing broader, low-frequency features) and larger radii representing finer, high-frequency details. Each index corresponds to one of these angles and scales, from which two notable observations are seen in this example and its first-order weights (column c).

First, there is a general increase in weight across all polar indices as the cyclone's structure intensifies, suggesting that the network places more importance on a wider range of scales and orientations as the storm becomes more organized. Second, certain groups of indices stand out as having stronger weights across timesteps. For instance, at $t-18$, the first-order features display relatively low variability, indicating fewer prominent edges or directional features, suggesting a relatively less developed storm structure at that time. By $t-9$, the largest scales (represented by the smallest radii) show the strongest response at an angle of $\pi/3$, which are visually orthogonal to the direction of the largest rainbands. At timestep $t$, there is a noticeable increase in both weight and variability, particularly at the smallest scales (largest radii), which likely corresponds to well-defined edges, such as those around the storm's eyewall and rainbands.




**(a)** IG **(b)** $\mathbf{A}^s$ **(c)** first-order **(d)** second-order

**Figure 2.12:** Feature visualizations from the tropical cyclone intensity data.

The second-order attention weights (column d) provide further granularity by subdividing the first-order quadrants. While much of the same behavior can be observed here, the weight of second-order coefficients show finer details about how the network responds to multi-scale interactions within the imagery. The stronger weights at smaller scales suggest the network is more attentive to subtle, high-frequency features as the cyclone becomes more intense.

In addition to visualizing the attention weights for a particular example, we also show the magnitude of attention weight for all coefficients across angles and channels for all test samples in Figure 2.13. These are averaged across scales to emphasize the angular importance corresponding to first- and second-order coefficients. The zeroth-order coefficients are essentially a lowpass filter, and do not have a rotation, but the others do. This allows us to see how attention, particularly toward certain angles, change with sample intensity of wind speed.

Generally, there is a decrease in the weight given to the zeroth-order coefficients as intensity increases (Figure 2.13a). This gradual decrease coincides with the development of well-structured storms that require greater detail to resolve a prediction. Subsequently, the weights of first- and second-order coefficients in Figures 2.13a and 2.13b show a steady increase in the magnitude of attention at certain angles as wind speeds

increase. This is more clearly seen in second-order weights and compliments the decreasing observation of what is seen in the zeroth-order.

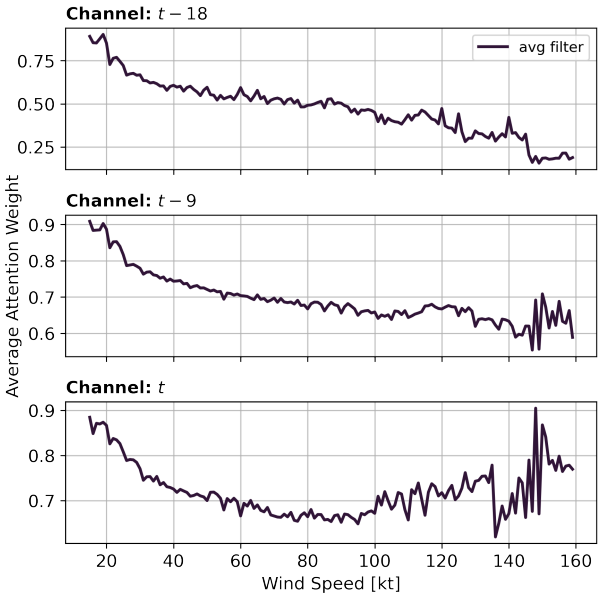
Interestingly, we observe a rapid drop followed by a variable increase around $\sim 135\,\mathrm{kn}$ across several channels, most notably in channel $t$, across coefficients shown in Figure 2.13. We speculate that this is due to the rapid intensification of the cyclones and structural changes occurring within this intensity range. Additionally, there is no clear evidence that one angle consistently performs better than another; when one angle is high, another tends to be low. This observation is appropriate given the circular nature of the events.

### 2.3.3 Short Range Lightning Prediction

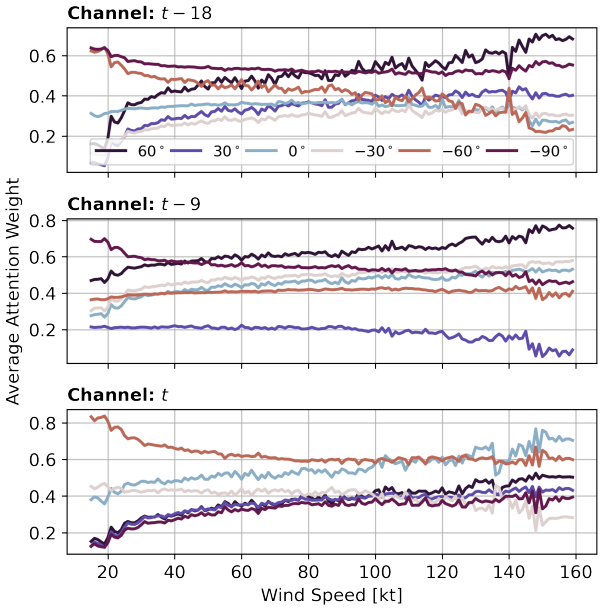
Accurate short-term predictions of lightning onset can help protect life and mitigate the economic impacts from disrupted outdoor work and natural fires by updating people on when to seek shelter and the persistence of lightning events. The AI for Earth System Science Hackathon [61] opens this challenge with data from GOES-16 ABI and aggregate lightning flash counts, lagged by one hour, from the Geostationary Lightning Mapper (GLM) [62]. The input channels include the following four water vapor bands: upper-level troposphere (band-8 or $6.2\,\mu\mathrm{m}$), mid-level troposphere (band-9 or $6.9\,\mu\mathrm{m}$), low-level troposphere (band-10 or $7.3\,\mu\mathrm{m}$), and longwave (band-14 or $11.2\,\mu\mathrm{m}$). The target flash counts are converted to binary labels and used for predicting if lightning is present in the previous hour from the locations captured by satellite.

The data consists of $32 \times 32$ image patches (for each band) across the Continental United States (between latitudes $29.09°$ and $48.97°$ and longitudes $-97.77°$ and $-82.53°$) at $20\,\mathrm{min}$ intervals from 2019-03-02 through 2019-10-01. We perform bilinear interpolation to each band, scaling the inputs to $64 \times 64$ for more flexible spatial attention features. This is done because the scattering transform yields coefficients of resolution $W/2^J \times H/2^J$ (i.e., $8 \times 8$) and resolutions too small will lose detail. The brightness temperatures, measured in kelvins, are min-max normalized to $[0, 1]$ using the statistics of the training data. In total there are $212{,}604$ training, $212{,}604$ validation, and $199{,}157$ test samples. Figure 2.14 displays an example input sample with each stacked channel-wise bands.

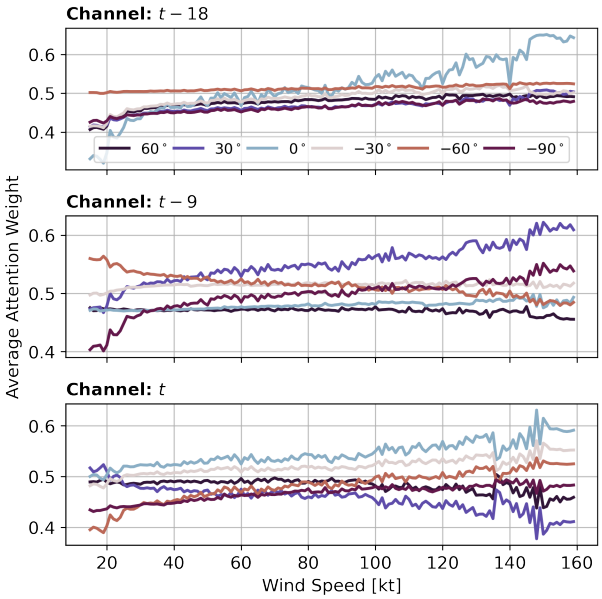
Flash counts from the GLM have a strong positively skewed distribution (i.e., $4.47 \pm 18.22$) across all training samples. When converting to binary labels, where true when flash counts are greater than zero, we get a better distribution of targets with a slight class imbalance of $63.49\%$ training samples having lightning. When subsampling the training data to have $n$ total samples ranging from $1{,}000$ to $212{,}604$ (Table 2.1), we reduce bias by maintaining this class distribution in sampling.




**(a)** Zeroth-order coefficients




**(b)** First-order coefficients




**(c)** Second-order coefficients

**Figure 2.13:** Attention weight of scattering coefficients versus wind speed averaged over wavelet scales, and separated across angles and the three temporal channels ($t-18, t-9$, and $t$). Zeroth-order coefficients **(a)** have no angles, whereas first-order **(b)** and second-order **(c)** have $L = 6$ angles between $[0, \pi)$, reflected about $0°$.

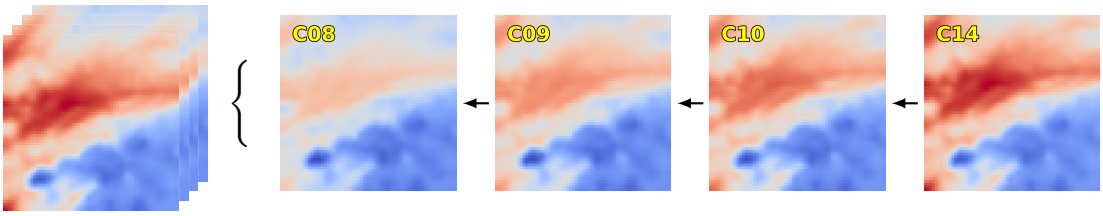



**Figure 2.14:** Input data of size $(1 \times 4 \times 64 \times 64)$ with stacked channel-wise brightness temperatures (annotated in yellow). The target variable is a binary label indicating the presence of lightning over the previous hour.

### Main Findings

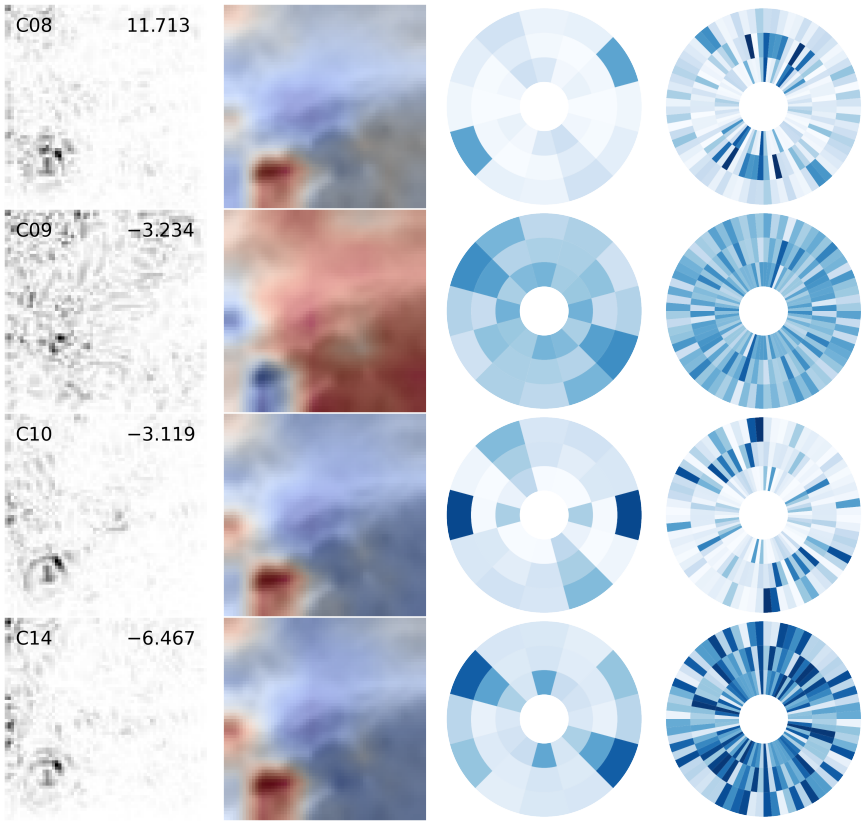
To the best of our knowledge there are no public benchmarks of this dataset. We therefore make comparisons with only the aforementioned models listed in Table 2.1. We also conduct experiments in manner similar to Section 2.3.2, specifically, comparing architectures with a change in training set size and exploring model interpretations.

The scattering network shows minimal sensitivity to sample size, with only a $4.42\%$ decrease in classification accuracy when trained on just $0.47\%$ of the data. In contrast, ResNet18, which has a marginal $0.05\%$ higher accuracy than the scattering network when trained on the full dataset, suffers a $12.36\%$ drop in accuracy when limited to $n = 1000$ training samples. Furthermore, for the same $n = 1000$ samples, the scattering network outperforms the convolutional network by $7.77\%$. These findings are summarized in Table 2.1. In summary, we find the scattering network excels in low-data scenarios, with very marginal loss in accuracy at drastically low sample sizes. However, relative to ResNet18, there are diminishing returns as the sample size increases to very-high data regimes.

### Network Interpretations

Figure 2.15 displays the attention features for a particular convective storm (the same one in Figure 2.14), where the presence of lightning flashes were correctly classified by the scattering network. In the raw imagery, moist convection is clearly visible, characterized by cold brightness temperatures at the cloud top. This cold region (in the lower left) results from clouds reaching higher altitudes, where atmospheric temperatures are significantly colder.

The spatial feature map $\mathbf{A}^s$ (column b) highlights that for channels 8, 10, and 14, there is greater attention (in red) focused on this cold spot. This aligns with our understanding of convective storms and their connection to lightning. Channel 9, however, shows an inverse pattern, with higher attention weights surrounding the cold spot. This distinction between channels suggests that the network is capturing complementary information




**(a)** IG **(b)** $\mathbf{A}^s$ **(c)** first-order **(d)** second-order

**Figure 2.15:** Feature visualizations from the short-term lightning prediction data.

across different spectral bands. Notably, the separation of attention is more local than just the cold or warmer regions; it specifically isolates the coldest, convective spot. This highlights our network's ability to dynamically localize relevant atmospheric features at inference time that are meteorologically important for a prediction.

The channel attention weights $\mathbf{A}^c$, corresponding to the first- and second-order scattering coefficients, display a highly sparse structure, meaning the network relies on only a few dominant orientations and scales to make its predictions. Specifically, channels 8, 10, and 14 show higher weights at larger radii, corresponding to finer spatial scales and high-frequency details. These weights peak at a single orientation that varies among channels, indicating that the network is sensitive to different directional components in each spectral band. The second-order scattering coefficients (column d) further reveal how the network captures interactions between these fine details across multiple scales.

### 2.3.4 Feature Visualizations

There are three prominent methods to visualize the inner computations of the scattering network. These methods, illustrated in Figures 2.12 and 2.15, provide insights into the model's use of spatial and channel attention features. The following paragraphs describe these in more detail:

**Spatial Attention Features** The separation of attention modules yield a spatial attention feature map, $\mathbf{A}^s$, for each input channel in the data. This map can be visualized via a bilinear upsampling from $W/2^J \times H/2^J$ to the original input resolution $W \times H$. Superimposing this scaled map on the original data highlights the spatial regions that are most attended to. The resolution and visual detail of the feature map depend on the scattering transform's scale, $J$. A larger $J$ results in a smaller feature map, while a smaller $J$ preserve more detail and fewer interpolation steps.

**Channel Attention Features** First-order coefficients are represented by a polar radius, which is inversely proportional to the scale $2^{j_1}$ of the wavelet $\psi_{\lambda_1}$, with the corresponding angle representing the rotation $r_1$. This approximates the frequency bandwidth of the Fourier transform $\hat{\psi}_{\lambda_1}$. Each quadrant is thus indexed by $(r_1, j_1)$. For second-order coefficients, each first-order quadrant $(r_1, j_1)$ is further subdivided along the polar radius for scales $j_1 < j_2 < J$ and for all $L$ angles, indexed as $(r_1, r_2, j_1, j_2)$. In both cases, we use the central symmetry of the wavelets, with angles in $[0, \pi)$, to illustrate them on a unit circle. Importantly, instead of visualizing individual scattering coefficients, we display the normalized channel attention value of $\mathbf{A}^c$ corresponding to each index.

**Gradient Based Methods** The scattering transform is differentiable, and thus, allows for evaluations of post hoc explainability methods. We demonstrate an example of how gradients can be computed with respect to an individual input pixel by computing integrated gradients for our example input. Following the work of [63], we establish a baseline of all zeros and compute importance scores for each pixel in the input. While we show integrated gradients as an example, alternative post hoc explainability methods, e.g., GradCAM, layer-wise relevance propagation, Shapley values, etc., could be used to evaluate this network.

## 2.4 Discussion

The proposed WaveNet model is a step toward interpretable-by-design networks, designed to learn dominate frequencies within the data. With very few trainable parameters, the network effectively reduces

the complexity of the convolutional filter to learn just two key parameters: frequency, $f$, and width, $w$. We highlight interpretability here as with a trained model, we can inspect the wavelet parameters after training to know exactly what features are being extracted from the data. We demonstrate the efficacy of our model on a simplified dataset, showing how the network successfully fits to the underlying frequencies. Notably, we observe grokking-like behavior in the network, where the frequencies converge rapidly after several training steps. Furthermore, WaveNet shows promise in gravity wave classification, outperforming traditional methods and achieving an accuracy of $88.32\%$ on a handcrafted satellite imagery dataset. In both cases, the network proves robust to noise and is capable of generalizing, where simpler, fully-connected networks tend to overfit.

The central contribution of WaveNet is in learning parameterized one-dimensional wavelet functions with backpropagation. While our focus is on the complex Morlet wavelet, this approach can be extended to other wavelet functions and network architectures to accommodate different data characteristics. This method is particularly effective for datasets with inherent periodicity, though its applicability may be broader with proper adjustments. However, a few limitations and challenges remain and should be considered when adapting this method.

The first challenge concerns the initialization of wavelet parameters. If the initial frequency is too far from the true frequency present in the data, the network may struggle to converge. To mitigate this, we find that freezing the width parameter while learning only the frequency can help stabilize training. Additionally, incorporating multiple filters to span the possible frequency range provides a brute-force alternative. Performing a prior analysis via a Fourier transform to identify the underlying frequencies could further hint at better initialization. The second challenge pertains to scaling the model for large datasets. Such data that consists of a wide range of frequencies may require a substantial number of wavelet filters, which could increase the complexity and computational cost. It is also not clear how a cascade of adaptive wavelet layers would perform.

In contrast to learning specialized wavelet functions, we learn how best to attend to their coefficients with our scattering network. The proposed separation scheme defines the most salient features and scattering coefficients on individual input channels and can easily be visualized to better understand the use of each channel. The result is a network that promotes interpretability and can easily be adapted to other computer vision or satellite-based tasks.

Our findings show that the scattering network, despite having fewer trainable parameters than a linear model, achieves $\sim 20\%$ lower error and better generalization than both standard convolutional networks and a subset of state-of-the-art vision models in a sample application of estimating tropical cyclone intensity. However, while we have strong results, we do not outperform large-scale ensemble methods. Encouragingly though, our network is highly effective under all data constraints, particularly when data is scarce. Similarly, the network demonstrates strong performance in predicting short-term lightning occurrence, with its greatest advantages seen with small sample sizes, but there are diminishing returns for very large sample sizes. Thus, highlighting our approach to be particularly well-suited for data-constrained applications involving imagery with multiple input channels.

Despite its effectiveness, there remains uncertainty regarding the contribution of individual components to the overall prediction. An ablation study would be beneficial to isolate which attention features drive the improved performance. This would provide a more cohesive understanding of how the scattering coefficients influence the model's decisions and help guide further development for specific use cases. Furthermore, we have observed that not every scattering coefficient is needed for an accurate prediction. Therefore, it would also be useful to evaluate other methods of combining high-level features, such as an informed method for selecting the top-$k$ weighted scattering coefficients after each attention module or other, simpler aggregate functions.

# Chapter 3

# Toward Sequential Attention for Computer Vision Tasks

In Chapter 2, we explored the use of wavelets and spatial attention to improve how local and global patterns in high-dimensional data are captured. While spatial attention provides a network with important regions and feature channels over an entire image, sequential attention iteratively builds up localized features from only a subset of image locations. At a high-level, we model "what" and "where" to look in an image, using only previously observed patches or glimpses, for predictive tasks.

More specifically, in this chapter, we propose a biologically-inspired model of sequential attention for image processing, named memory-based sequential attention (M-SAtt) that draws on the principles of the human visual system. We discuss and address the limitations of previous work, e.g., with recurrent-based sequential attention, and introduce a transformer-based memory module coupled with a hybrid reinforcement learning- and data likelihood estimation-based algorithm to learn a control strategy of where to look in a visual scene. Our approach improves not only task performance but also our insight into the decision-making process of the model. While we validate our approach on classical vision tasks, we also demonstrate its practical application in climate science, where identifying and focusing on particularly salient regions is crucial for understanding climate variability and its role as an indicator of climate change.

We begin by motivating the need for sequential attention and providing an overview of the cognitive theories and neural models of attention in Section 3.1. In Section 3.2, we detail our methodology and the specifics of our proposed architecture. We then demonstrate and discuss the results on classical vision tasks in Section 3.3, followed by adaptations applied to the climate domain in Section 3.4. Finally, we conclude with a summary of sequential attention in Section 3.5, reflecting on the implications for both the broader field of computer vision and its specific applications in atmospheric science.

Additional reading as it relates to this chapter can be found in the corresponding publication:

> Stock, J., & Anderson, C. (2023). *Memory-Based Sequential Attention*. In NeurIPS 2023 Workshop on Gaze Meets Machine Learning (PMLR), Dec, 2023.

## 3.1 Background and Motivation

The human visual system constantly receives a vast amount of data, with estimates ranging from $10^8$ and $10^9$ bits of information per second [64, 65]. Given the sheer volume of this input, it is crucial to have some mechanisms in place for filtering out extraneous or erroneous data to effectively process it in real-time. To accomplish this task, the visual system relies on advanced cognitive processes and forms of dynamic attention. Underlying this fundamental principle are evolved mechanisms for selection based on some notion of relevance.

The basis for many computational models of attention build on the pinnacle work of Treisman and Gelade [66], who proposed the "Feature Integration Theory". Intuitively, this theory suggests that attention can be directed towards visually distinct regions or features that stand out in comparison to their surroundings. The importance, relating to human perception, is that the entire visual scene is not processed at once. Rather, we selectively build an internal representation based on localized information. A given location or memory of previous locations may be informative for where to look next, where the total history of locations may influence scene interpretations.

Our proposed method takes inspiration from biological models of visual attention for describing global scene understanding with visual scanpaths or trajectories. To get a better intuition of computational models of attention, we first review these concepts as a cognitive process in Section 3.1.1, where the fundamental theories on visual control from psychology and neuroscience are introduced. Thereafter, we discuss neural models of attention and how they relate to this work in Section 3.1.2.

### 3.1.1 Visual Attention as a Biological Process

Classical studies in cognitive psychology and neuroscience isolate selective attentional control to follow two predominate theories in vision [67, 68]. The first is a *top-down* process (also called endogenous or goal-directed), whereby control is volitional and activated by the observer. This may use one's belief and 'internal' factors to control attention. On the contrary, a *bottom-up* (also called exogenous or stimulus-driven) theory suggests control is involuntarily driven by factors external to the observer. Stimuli that are physically salient due to their inherent properties relative to the surrounding environment are likely to capture attention. Here, the salience of a stimulus is defined by low-level visual characteristics, including modalities of color, intensity, orientation, and motion [69].

Both theories of attentional control are shown to work in tandem for selective attention, but these processes alone do not fully explain the range of phenomena related to attention. Only recently, however, the dichotomy of top-down and bottom-up control has been challenged with evidence for additional factors that control visual selection, such as *reward-based history* effects [68, 70]. The underlying idea is that past episodes of attentional selection can strongly influence current selection above and beyond top-down and bottom-up processing. This is particularly evident in studying the interactions between rewards and attention, where rewards can shape both perceptual and attentional processes, prioritizing certain stimuli and modifying spatial and temporal attentional selection.

The processes of top-down, bottom-up, and reward-based history also control both covert and overt processing of visual information [71, 68, 72]. *Covert attention* refers to the internal processing of visual information without any *saccadic eye-movements*, the rapid eye-movements that occur when shifting gaze between locations. This allows for parallel processing and quick interpretations of visual information. It is intuitively used to monitor the environment and guide eye-movements, allowing us to attend to a target without fixation. *Overt attention*, on the other hand, is associated with a fixation as eye-movements direct attention to different locations in the environment or visual scene. Attentional focus, therefore, occurs within the line of sight of the *fovea*, the central part of the retina responsible for sharp, central vision. Overt attention is an intentional and conscious process that allows us to focus on specific stimuli in the visual scene, often to gather more detailed information related to a given task.

### 3.1.2 Computational Models of Attention

As early as 1987, Koch and Ullman conceptualized a feed-forward model to aggregate salient features based on color, intensity, and orientation to compute a saliency map emphasizing conspicuous locations. A "winner-take-all" approach, based on inhibition of return [74], was then utilized to shift the focus of attention to the next salient region. This approach was later implemented and validated as a computational model for digital images [75, 51, 69]. Building on these early works, much effort has been devoted to modeling saliency maps and the development of attention for predictive vision tasks [64, 76]. These models often combine top-down and bottom-up processes, driven by internal computations within the model. Feed-forward methods, which we define as spatial and contextual attention mechanisms, serve as a precursor to sequential attention.

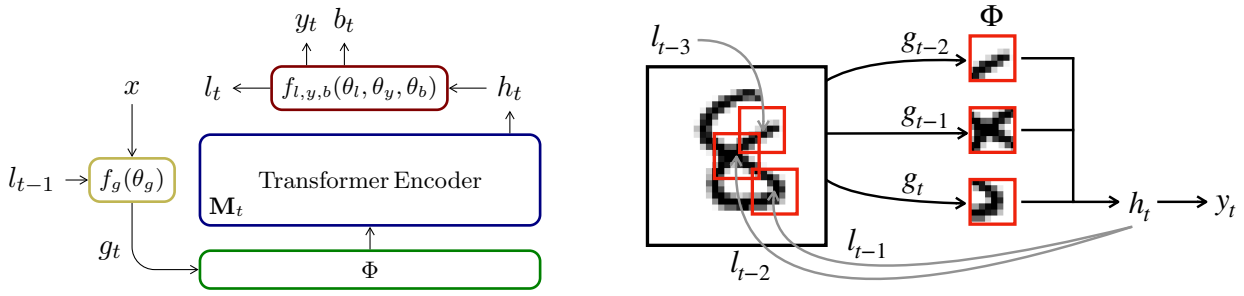
*Soft spatial* attention methods resolve salient features with a continuous-value mask. This can occur in the spatial domain [50, 52, 77] or as a special extension over channels [78, 52] as we saw in Chapter 2. In

contrast, *hard attention* localizes and crops selective regions to process for the relevant task. This can be done by learning a transformation over the input [79], with region proposals [80], or by learning a masking strategy [81, 82]. *Contextual attention* is inspired by the relationship between top-down and bottom-up visual cues that explain attentional deployment. In this context, the guidance of selection bias over *values* (sensory inputs) with attention pooling considers the interactions of a given *query* (top-down, volitional cue) and a set of *keys* (bottom-up, nonvolitional cues). This relationship is more commonly modeled with scaled dot-product attentional pooling, motivating the transformer architecture [83]. The natural extension of this concept to visual scenes was proposed in [84], where attention weights correspond to the contextual relationships of image patches. However, attention to the entire input can introduce irrelevant information and unnecessary computations.

In contrast to these static approaches, attention has also been studied as a sequential decision-making process [85–91]. Instead of identifying fixation zones based on bottom-up or top-down features, some studies, including this work, model attention as a sequential learning problem. From our background on cognitive processes (Section 3.1.1), we can model the reward-based history effects of visual control to guide the learning of task-relevant trajectories through reinforcement learning. The task of where to attend therefore becomes a sequential learning problem, requiring covert sampling of a sensory scene.

Modern works often build on the foundational work of Mnih et al. [86] and use recurrent models for sequential attention. However, these models face two significant limitations. First, as the task progresses, state representations are accumulated, leading to potential information loss as earlier states are compressed into a fixed-size memory, reducing the model's ability to retain fine-grained details from previous observations. Secondly, recurrent models lack the ability to dynamically reweigh glimpses, making it difficult to adjust the importance of previously attended information as new glimpses are sampled.

In this chapter, and to address these limitations, we replace the recurrence with a transformer encoder comprised of multiple self-attention heads. This is similar to the vision transformer introduced in [84], which processes patches spanning the entire visual scene, but our approach samples patches one at a time from continuous locations within the environment. This reduces sequence length while enabling dynamic, contextually driven attention to input features.




**(a)** Network Architecture **(b)** Flow Diagram

**Figure 3.1:** High-level architectural overview of our proposed model. A predicted location $l_{t-1}$ samples a glimpse, $g_t$, from an input image, $x$. The glimpse is added to a memory store, $\Phi$, where a masked transformer block computes a hidden state, $h_t$. The hidden state is used to emit the next location, $l_t$, a predicted class, $y_t$, and baseline estimate, $b_t$.

## 3.2 Memory-Based Sequential Attention

Instead of processing the entire image $x$ at once, we sample smaller patches, or glimpses, for sequential decision-making. Our proposed model illustrated in Figure 3.1 stores the memory of previously visited glimpses from a trajectory, and contextualizes their relationships to learn a strategy of "where to look". This approach uses reinforcement learning and data likelihood estimation to optimize for classification tasks (and later for regression in Section 3.4).

### 3.2.1 Preliminaries

The recurrent model of visual attention (RAM) [86] relies on a recurrent neural network to sample glimpses. At each time step $t$, the agent observes a glimpse of the environment from a particular continuous-valued location, $l_t = (i_t, j_t)$, and accumulates this over time to determine a location for the next step. A scalar reward is emitted at each step, where in a classification setting, a positive value is given if the class is correctly predicted. The goal is, therefore, to select a sequence of observations from the environment that maximize the total cumulative reward.

In this chapter, we replace the recurrent network state representation with a modified vision transformer, as detailed in Section 3.2.2. However, we leverage from RAM the following components:

**Glimpse Network** The previous location $l_{t-1}$ is used to sample a retina-like representation (or glimpse) $\rho(x, l_{t-1})$ from the full image, $x$, providing the agent only a partial view of the scene at time $t$. An initial location is set as $l_0 = (0, 0)$ or drawn randomly within some range between $[-1, 1]$, where $(-1, -1)$ is the

top left and $(1, 1)$ is the bottom right. The resolution of a glimpse is defined by the number of scales, $s$, composing high- and progressively lower-resolution regions around the location, stacked as separate channels.

We use nonlinear, fully-connected layers to extract the embedding of a given glimpse and model "what" it represents. We also use the transformed location to capture "where" the glimpse is located. We then combine the output of these two models, capturing the "what-where" combination, through a subsequent nonlinear transform to create a glimpse feature vector represented by $g_t = f_g(x, l_{t-1}; \theta_g)$.

**Location Network** At every timestep, a location, $l_t$, is emitted by the location network, $f_l(h_t; \theta_l)$, using the hidden state, $h_t$, of the model as calculated in Section 3.2.2. The location is stochastically sampled from a parameterized Gaussian distribution with fixed variance as $l_t \sim p(\cdot|f_l(h_t; \theta_l))$, where the mean of the $i, j$ coordinates are estimated by the location network. In the context of reinforcement learning, we sample values from the location policy, where the policy function maps the current observation of the environment (in this case, the hidden state, $h_t$) to the action to be taken by the agent (the continuous-valued location, $l_t$). During training, the log probabilities of the sampled locations are used to update the network. Section 3.4.2 will also discuss changes to further improve exploration.

**Classification/Output Network** Similar to the location network, we use the hidden state, $h_t$, to predict a class by passing it through an additional network that outputs the softmax over a set of possible classes. Specifically, we represent this as $y_t = \arg\max_c p(c \mid f_y(h_t; \theta_y))$ from which a class is selected. In Section 3.4.2, we will detail the modifications needed for regression.

### 3.2.2 Contextual Attention over Memory

Discovering the next best location of where to look or interpreting what has been seen over some sequence is a complex task. It is not always the case that every observed location is relevant. However, the relationship between different regions, or even from a single location, may be informative for a given task. This assumes ample exploration of the visual scene with memory of what has already been observed.

Using the glimpses, or multi-resolution patches, that we sample from the visual scene (as described in Section 3.2.1), we populate a *memory store*, $\Phi = \{g_0, \ldots, g_k\}$, that buffers the history of all observed glimpses in the trajectory. Treating each glimpse as a token, we use self-attention to contextualize their relationship and highlight important locations to compute an embedding, $h_t \in \mathbb{R}^d$, that is used to emit the next location or class prediction. A traditional vision transformer [84] will partition an image, $x \in \mathbb{R}^{c \times h \times w}$,

into $n$ equally divisible patches of size $p$ such that $n = (w/p) \cdot (h/p)$ with dimension $d = p^2 c$. However, $\Phi$ has a sequence length $k \ll n$ from sequentially concatenated glimpses with an embedding dimension $d$ from the glimpse network.

A standard sine-cosine positional encoding is added to the set of tokens in memory and padded with zero-valued vectors to ensure a fixed length $k$ (rotary position embeddings [92] could be explored as an alternative). Let matrix $\mathbf{X} \in \mathbb{R}^{k \times d}$ be the new row-wise concatenation of the tokens. We begin to compute $h_t$ with a single transformer block composed of multi-head self-attention (MSA) and a residual point-wise fully-connected network (FCN) as,

$$\mathbf{Z} = \text{FCN}(\text{MSA}(\mathbf{X})) \quad \text{such that} \tag{3.1}$$

$$\text{MSA}(\mathbf{X}) = [\mathbf{O}_1, \mathbf{O}_2, \ldots, \mathbf{O}_h]\mathbf{W}^{\mathbf{O}}, \tag{3.2}$$

where $h$ is the number of heads, $\mathbf{W}^{\mathbf{O}} \in \mathbb{R}^{hv \times d}$ are trainable weights, $[\cdot]$ is the column-wise concatenation, and $\mathbf{O}_i \in \mathbb{R}^{k \times v}$ is the output of the $i$-th attention head with latent dimension $v < d$. We introduce a mask $\mathbf{M} \in \mathbb{R}^{k \times k}$ to ignore the padded and yet to be observed locations and compute each head as,

$$\mathbf{O}_i = \mathbf{A}_i \mathbf{V}_i \quad \text{such that} \tag{3.3}$$

$$\mathbf{A}_i = \text{softmax}\big((\mathbf{Q}_i \mathbf{K}_i^\top + \mathbf{M})/\sqrt{d}\big) \in \mathbb{R}^{k \times k}. \tag{3.4}$$

This mask effectively pushes the attention weights of padded locations, across all batches similar to causal masking, in the softmax toward zero with,

$$\mathbf{M}_{*,j} = \begin{cases} -\infty & \text{if } j \geq t \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

The queries, $\mathbf{Q}_i$, keys, $\mathbf{K}_i$, and values, $\mathbf{V}_i$ are found via a linear projection of $\mathbf{X}$ by,

$$\mathbf{Q}_i = \mathbf{X}\mathbf{W}_i^Q, \quad \mathbf{K}_i = \mathbf{X}\mathbf{W}_i^K, \quad \mathbf{V}_i = \mathbf{X}\mathbf{W}_i^V, \tag{3.6}$$

with trainable weight matrices $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V \in \mathbb{R}^{d \times v}$.

The FCN is a two layer residual network separated by the ReLU activation, $\delta$, and dropout ($p = 0.2$) that takes as input the layer normalized (LN) residual output from above, defined by $\bar{\mathbf{X}} = \text{LN}(\mathbf{X} + \text{MSA}(\mathbf{X}))$. We then compute this as,

$$\text{FCN}(\bar{\mathbf{X}}) = \text{LN}(\bar{\mathbf{X}} + \delta(\bar{\mathbf{X}}\mathbf{W}^R)\mathbf{W}^S), \tag{3.7}$$

where $\mathbf{W}^R \in \mathbb{R}^{d \times m}$ and $\mathbf{W}^S \in \mathbb{R}^{m \times d}$ such that $m > d$. The output is reshaped from $k \times d \to kd$ and linearly projected into $h_t = \mathbf{Z}\mathbf{W}^{\mathbf{z}}$ with $\mathbf{W}^{\mathbf{z}} \in \mathbb{R}^{dk \times d}$ (see Equation (3.1)). This hidden state is used for the subsequent network components and repeats at every timestep with an updated $\Phi$ until termination.

Note that the effective padding and masking steps could be left without (in Equation (3.4)) by allowing $\Phi$ to have a dynamic sequence length. The output of scaled-dot product attention will also have a variable length. Thus, we can compute the mean over the sequences to obtain a $d$-dimensional vector, yielding $h_t$. However, we find this approach to be insufficient, as empirical evidence supports the importance of reweighing glimpse extrema in computing $h_t$.

### 3.2.3 Training Procedure

We view the problem of "where to look" as a control problem, or Partially Observed Markov Decision Process (POMDP), where the next transition only depends on the current state (i.e., memory of previous glimpses) and action (i.e., continuous-valued location). The objective, as a reinforcement learning problem, is to learn a strategy or sequence of actions that maximizes the cumulative reinforcements along a trajectory. At each timestep an agent selects an action, $a_t \in \mathcal{A}$, in the current state, $s_t$, according to its policy. A scalar reward $r(s_t, a_t)$ is received and then transitions to the next state $s_{t+1}$ following the probability $s_{t+1} \sim p(\cdot|s_t, a_t)$.

Consider a stochastic policy $\pi_\theta$, parameterized by a neural network, such that we aim to maximize the expected return $J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[R(\tau)]$. We assume an episodic environment with $\tau = (s_0, a_0, \ldots, s_{T+1})$ where we can estimate the expectation with a sample mean given a set of trajectories, $\mathcal{D} = \{\tau_i\}_{i=1,\ldots,N}$. By the policy gradient theorem, and shown by Williams [93], we arrive at an approximation to derive the analytical gradient,

$$\nabla_\theta J(\pi_\theta) \approx \frac{1}{|\mathcal{D}|} \sum_{\tau \in \mathcal{D}} \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)\ (R(\tau)_t - b(s_t)). \tag{3.8}$$

In this approximation, we include a baseline that does not depend on the action, e.g., an estimate of the value function $b(s_t) = \mathbb{E}_\pi[R(\tau)_t]$, to reduce variance and improve convergence. We emit this baseline along with the next location $l_t$. The gradient of this expected return or learning rule (Equation (3.8)) is also referred to as *REINFORCE with baseline*

Following Mnih et al. [86], only the location network is trained by maximizing $J(\pi_\theta)$. In doing so, the gradient information does not flow to any other network components. We train a baseline network, for use with optimizing the locations, by minimizing the mean squared error with the rewards at each step, $\mathcal{L}_b = \sum_{t=0}^{T}(b_t - r_t)^2$, again, restricting gradient flow to the rest of the model. All other network components are updated to minimize the cross entropy loss $\mathcal{L}_y = -\sum_{c=1}^{M} t_{i,c}\log(p_{i,c})$, with ground truth class labels, unless if used for regression (see Section 3.4.2).

### 3.2.4 Comparison to a Vision Transformer

Self-attention has a complexity of $\mathcal{O}(n^2 \cdot d)$ that is quadratic in the sequence length, $n$. Standard vision transformers assume the image size is divisible by the patch size, $p$, to compute the constant sequence length. The computational and memory cost, therefore, is especially noticeable for a large image, as $p$ decreases, and as the number of attention layers increase. Pruning irrelevant tokens within hidden layers, e.g., with sparse transformers that use additional prediction networks [94, 95] or scoring functions [96–98], is one such way to reduce complexity. However, these approaches still operate on the full-sized image as input and the nonlinear combination of tokens over layers makes it challenging to interpret their true representation.

In this chapter, we learn a control strategy to sequentially sample $k \ll n$ ideal patch locations from the input, i.e., selecting relevant patches rather than removing them. This improves interpretability of relevant tokens and more closely aligns our model to the biological visual system. Furthermore, we do not need to assume the number of patches is a product of the image size, thus, allowing our method to scale to any size input.

Additionally, vision transformers do not have any inductive biases of translation or scale invariance. This means that objects that have a relative change in their position will result in a different response. While these aspects can be learned with ample data or through augmentations, it can still hinder performance. This is evident when comparing a convolutional network, that is translation invariant, to the vision transformer on cluttered MNIST (see Section 3.3.2 and Table 3.1b). In contrast, our location policy samples a glimpse along a trajectory that is optimized for the task. This inherently results in features that are invariant to translation by

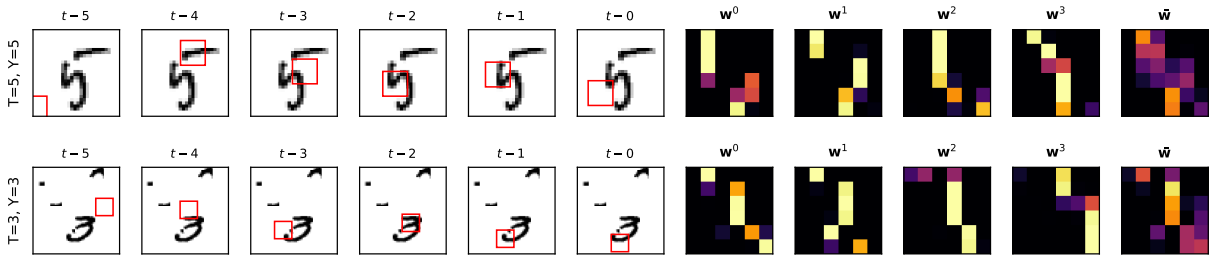



**Figure 3.2:** The learned policy of our best models and distribution of attention weights. An example from MNIST (top) and from cluttered MNIST (bottom) dataset. Columns 1-6 illustrate the trajectory of individual glimpses in red from the first timestep on the left up to the point of prediction. Columns 7-11 visualize the distribution of self-attention weights for each attention head with the associated mean in the last column.

learning to ignore irrelevant locations. As such, our method benefits from the properties of vision transformer models while also being translation invariant with a shorter sequence length.

## 3.3 Experiments on Classical Vision Tasks

We evaluate our method for classification on MNIST and cluttered MNIST datasets. Using a predefined number of glimpses, we output our final class prediction at the last step. During training, a reward of $1$ is assigned if the target class is correctly predicted, and $0$ otherwise, assigning all previous steps this value. Results are compared to differing network architectures of increased complexity. As a baseline we include a fully-connected and convolutional ReLU network with unit and filter sizes defined in Table 3.1. We also compare results with a standard vision transformer [84], which has a fixed patch size that equally subdivides an image over the entire input space.

Our comparison and implementation of RAM follows the architecture as described by Mnih et al. [86]. However, as there were no reported hyperparameters, our results slightly differ from the original. In our approach, we maintain hyperparameter consistency with replacement of the recurrent network for our memory-based transformer. A complete list of parameter values (e.g., optimizer, training epochs, etc.) are saved and can be found in our code repository. In the following experiments we use a single transformer block and vary the number of self-attention heads between $1$ and $4$.

All models are implemented in PyTorch with experiments conducted on a single node with an NVIDIA GeForce RTX 3090 (24GB), Intel i9-11900F (2.50GHz), and 128GB memory. All models have a similar number of trainable parameters.

### 3.3.1 MNIST Classification

We use the MNIST dataset, of size $28 \times 28$, to demonstrate the effectiveness of our approach. Data are partitioned into training ($50{,}000$), validation ($10{,}000$), and test ($10{,}000$). The validation data is only used for hyperparameter tuning. Evaluation results are summarized in Table 3.1a, where the top-$1$ classification error (lower is better) is reported on the test set. We use a sequence length of $k = 6$ with a glimpse size of $8 \times 8$ and a scale $s = 1$ for our approach and with RAM.

As we increase the number of self-attention heads from $1$ to $4$, we notice a steady decrease in error that eventually plateaus around $1.00\%$. However, we find that our performance is competitive to the implementation of RAM. Interestingly, we achieve a lower error than that reported in their original paper (see Mnih et al. [86], Table 1[2]). We speculate this is as a result of hyperparameter tuning.

The vision transformer uses a patch size of $7 \times 7$ that equally spans the image domain. The number of heads are equal to our best-performing model with the same linear transform and feed-forward embedding dimensions. We find the performance is worse than all of our tested models, suggesting that our learned policy is able to identify the most task-relevant glimpses from a shorter sequence length.

The top row of Figure 3.2 shows a sample from our best-performing model with a trajectory learned by the policy. A red box surrounds the glimpse locations from each timestep as it moves throughout the image. The model only observes the cropped information inside of this box and all outside information is discarded. To the right are the distribution of attention weights for each head with their associated mean. Attention weights are associated with the most conspicuous location. We find locations over the digit with the highest attention weights, whereas uninformative locations have low weight. This result validates our intuition as to what locations represent a digit. Additional details for interpreting these weights are made in Section 3.3.4, with more examples in Section 3.3.4.

### 3.3.2 Cluttered and Translated MNIST

To further test our approach we evaluate results on the cluttered MNIST dataset [99]. This data contains an MNIST digit that is randomly translated within a $60 \times 60$ canvas. Four different $8 \times 8$ subpatches sampled from other random digits are added at random locations. The presence of clutter as a form of noise make the task particularly challenging. Compared to the centered MNIST data, accurate predictions are more

[2]Officially reported top-1 classification error of $1.07\%$ in [86].

**Table 3.1:** Classification results as percent of test samples incorrectly classified, where S is the glimpse scale, H is the number of attention heads, and K is the number of sequential glimpses.

| MODEL | ERROR |
|---|---|
| FC, 2 LAYERS [256, 256] | 2.20 |
| CNN, 2 LAYERS [16, 32] | 1.17 |
| VIT, $7 \times 7$, 4H | 3.48 |
| RAM, 6 K, $8 \times 8$, 1 S | **0.94** |
| OURS, 6 K, $8 \times 8$, 1 S, 1 H | 1.29 |
| OURS, 6 K, $8 \times 8$, 1 S, 2 H | 1.11 |
| OURS, 6 K, $8 \times 8$, 1 S, 4 H | **1.05** |

**(a)** MNIST

| MODEL | ERROR |
|---|---|
| FC, 2 LAYERS [256, 256] | 56.82 |
| CNN, 4 LAYERS [8, 16, 32, 64] | 6.71 |
| VIT, $12 \times 12$, 4H | 29.48 |
| RAM, 6 K, $12 \times 12$, 3 S | 6.43 |
| OURS, 6 K, $12 \times 12$, 3 S, 1 H | 7.89 |
| OURS, 6 K, $12 \times 12$, 3 S, 2 H | 7.47 |
| OURS, 6 K, $12 \times 12$, 3 S, 4 H | **6.20** |

**(b)** Cluttered & Translated MNIST

**Table 3.2:** Policy error (mean $\pm$ std) when permuting the starting location in our best model, then following the learned policy. The last line is a stochastic policy that samples a random action (location) at each step.

| POSITION | MNIST | CLUTTERED |
|---|---|---|
| RANDOM | $\mathbf{1.12 \pm 0.03}$ | $6.76 \pm 0.07$ |
| TOP-MIDDLE | $1.13 \pm 0.05$ | $7.13 \pm 0.11$ |
| TOP-LEFT | $1.16 \pm 0.03$ | $7.02 \pm 0.22$ |
| CENTER | $1.20 \pm 0.05$ | $\mathbf{6.36 \pm 0.19}$ |
| BOTTOM-MIDDLE | $1.20 \pm 0.07$ | $7.32 \pm 0.15$ |
| BOTTOM-RIGHT | $1.12 \pm 0.05$ | $6.74 \pm 0.18$ |
| RANDOM POLICY | $29.49 \pm 0.28$ | $25.66 \pm 0.03$ |

dependent on a model that is invariant to translation and can learn to ignore the clutter that is not task-relevant. Data are partitioned into training (50,000), validation (10,000), and test (10,000).

Table 3.1b shows the classification results from the different architectures. For our model and our RAM implementation, we use a glimpse size of $12 \times 12$ with $s = 3$ to capture multi-resolution features over a sequence length of $k = 6$. Similarly, the vision transformer uses the same patch size with four attention heads. The benefits of our model is especially noticed when comparing to the vision transformer. RAM and our model achieve less than $7\%$ error, while the vision transformer achieves about $29.5\%$ error with the same number of attention heads.

Results show the fully-connected network performs worst, whereas the convolutional network performs significantly better with its inductive biases of translation invariance. However, our memory-based attention model shows a slight advantage as we learn a policy to avoid the clutter and focus on the different parts of the digit. We outperform RAM with the advantage of having attention weights that bring insight to how these glimpses from memory relate to each other.

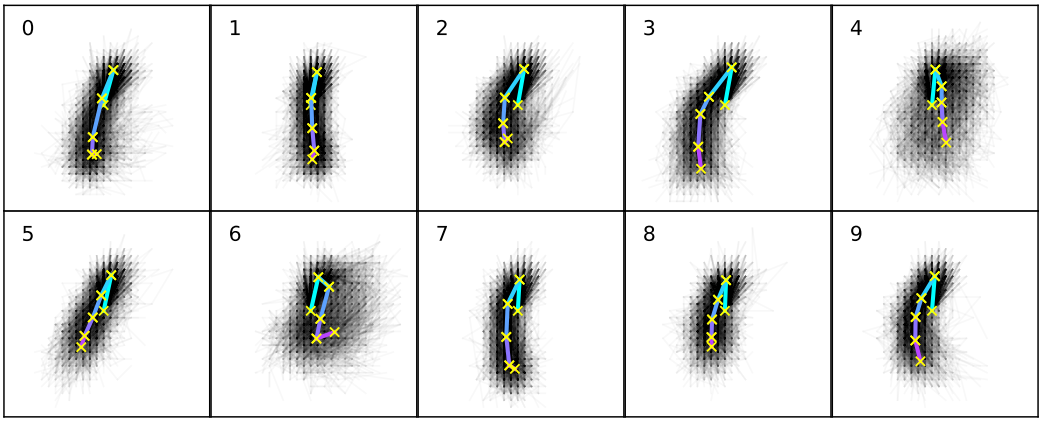



**Figure 3.3:** Class specific trajectories of all MNIST test samples in black with the mean trajectory overlaid. The mean glimpse locations, ×, all have a centered initial position, then progress following the path from cyan to purple.

As with the MNIST digits, we show an example trajectory and our model's attention weights in the bottom row of Figure 3.2. We find that for the last step class prediction, the glimpses from memory attend primarily to those that are directly focused on the digit, illustrating how our policy avoids the clutter while exploring the visual scene. Alternatively, indices with near zero values are found where the high-resolution glimpse is not focused directly on the digit.

### 3.3.3 Location Permutations

To evaluate model robustness we make inference with our best-performing models and compare the results to a random policy. The results for each dataset are shown in Table 3.2, where the random policy has locations sampled from a uniform distribution at each step. Each of the six starting positions tested herein follow the learned policy, but vary in where the first glimpse is sampled. This effectively shows the learned policy, with $\sim 20\%$ lower error, optimally selects locations for the given task.

There are slight variations in classification accuracy for different starting locations. The change is minimal with MNIST, but with the cluttered and translated MNIST dataset, there is a considerable influence. Notably, we find lower error when the initial glimpse is sampled from the center of the image, and a higher error when sampling from the periphery. We speculate this to be as a result of there being greater coverage of the multi-resolution glimpse, with periphery information, that can more accurately resolves the high-resolution features in subsequent steps.

In Figure 3.3, we show how inference trajectories vary among MNIST test samples for each class independently. By taking the trajectory mean, across all samples in each class, we glean insights into the global path and policy they follow. The following observations are based on this mean. Generally, for each class, the second glimpse is made at the top of each digit with a trend, scanning down the image that follows. The differences between each class are evident and supported by the structure of the digit. Take, for example, class '6', where the final glimpse moves right to sample the commonly enclosed circle of the digit. Without this, the sampled glimpses are similar to a '1'. Class '3' trajectories scan further left, presumably to delineate between class '8'. Lastly, class '4' trajectories are further spread around the top of the digit.

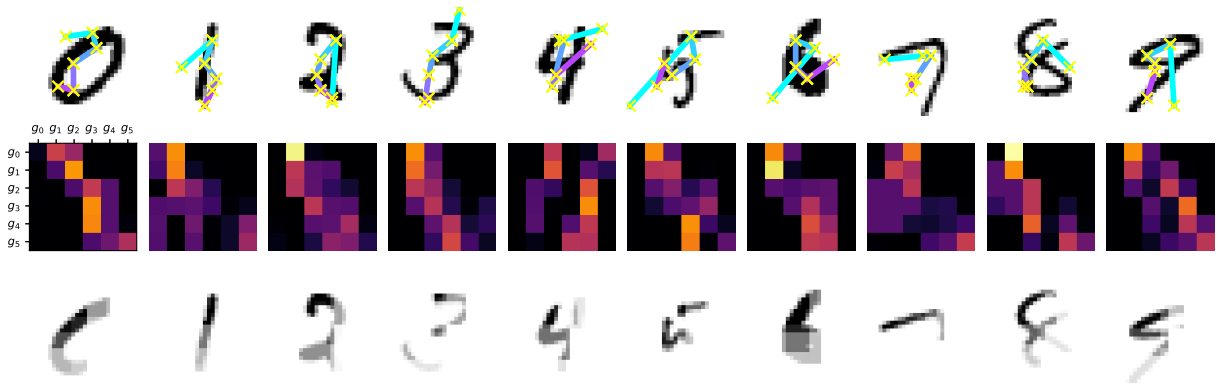



**Figure 3.4:** Sampled glimpse locations, $\times$, from within a trajectory (top), starting in cyan and ending in purple. Mean self-attention weights computed over all heads for glimpses $g_{0-5}$, where yellow is 1 and black is 0 (middle). Re-weighted glimpse locations found by the accumulated attention weights (bottom).

### 3.3.4 Network Interpretations

**Glimpse Trajectories** Learning a strategy of where to look results in an interpretable process of decision-making, allowing for the assessment of influence from individual glimpses for a given task. Manual review of these locations can help one reason about how the network arrives at the prediction. This is as a result of reducing the problem to a subset of locations from the entire scene, ignoring extraneous or erroneous data. However, it is also important to consider how the network uses these locations beyond human intuition. In our approach, we provide additional transparency by contextualizing over the observed glimpses in memory. By inspecting the distribution of self-attention weights we can glean insights to how these locations are attended to.

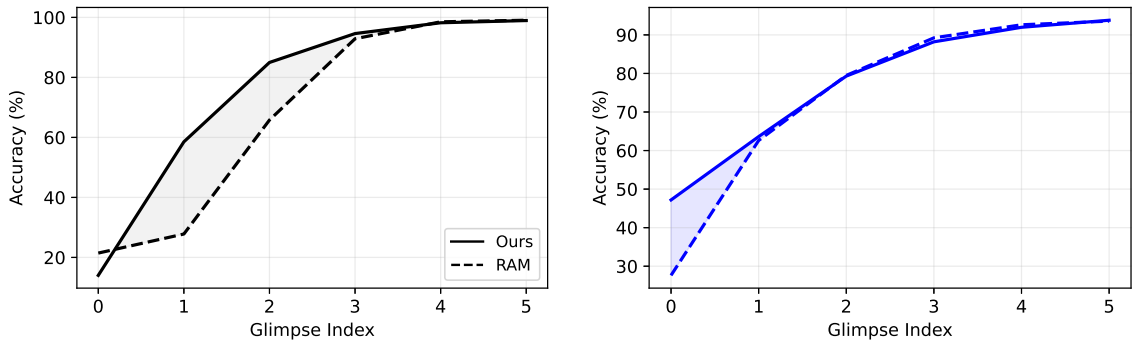
Figure 3.4 shows the final step trajectory for different MNIST digits. The mean over attention heads is computed to capture comprehensive relevance. Along each axis are weights corresponding to the relationship of each glimpse, such that the diagonal represents how a glimpse attends to itself. Interestingly, we find that glimpse locations that are task irrelevant, i.e., zero-valued locations or at locations with clutter, have little to no positional significance in the sequence. In contrast, the locations centered on the most conspicuous locations of the digit are largely attended to.

To exemplify this intuition, we accumulate the mean attention weights of each column to weigh each individual glimpse location. The last row of Figure 3.4 shows this result, where the target regions of increased clarity correspond to the most attended locations in the trajectory. Intuitively, this re-weighting highlights the features most distinctive to each digit.

Furthermore, we show visualizations of the learned policy for our best models and distribution of attention weights with examples from each class in the MNIST (Figure 3.6) and cluttered MNIST (Figure 3.7) test datasets. Columns 1-6 illustrate the trajectory of individual glimpses, which are cropped from the original image, in red from the first timestep on the left up to the point of prediction. The class prediction $y$ is labeled next to the target image $t$. Note that some samples in Figure 3.7 are incorrectly classified, and evidently have a poor trajectory. Columns 7-11 visualize the distribution of self-attention weights from the $4$ attention heads and their associated mean.

**Timestep Performance** Additional insights on the impact of subsequent glimpses are made by emitting the class prediction after every timestep. This is helpful to understand more generally how the inclusion of each glimpse contribute to the improvement in model performance. Figure 3.5 illustrates this result for our model as it compares to RAM on both datasets. With MNIST (Figure 3.5a), we find a $31\%$ and $19\%$ average increase in accuracy after glimpses $g_1$ and $g_2$ are observed, respectively. For the cluttered dataset (Figure 3.5b), we find a $20\%$ increase in accuracy in the first glimpse and marginally higher accuracy following. These findings indicate that our location policy can quickly identify locations that our model can contextualize over for classification. Furthermore, displaying the general and steady increase in performance that eventually plateaus with more glimpses.

**Input Units Weights** The glimpse network learns the "what" features from a glimpse of the input image. A fully-connected layer takes as input a glimpse before combining it with the "where" features. Our best




**(a)** MNIST **(b)** Cluttered

**Figure 3.5:** Test accuracy emitted after each glimpse of our approach (M-SAtt) as it compares to RAM. The shaded region emphasizes the performance improvements of early glimpses

model on the MNIST data uses $256$ input units to represent this fully-connected layer. After we have trained the network, and made updates to the weights in this layer, we can visualize each unit separately. These visualizations are made to better understand, to some degree, how the network represents a glimpse and arrives at its prediction.

In no particular order, we show these units in Figure 3.8. These units operate on the unstandardized glimpse of MNIST digits with intensity values between $[0, 1]$. The majority of weights are positive (as shown in red), but reveal some interesting patterns. That is, there are strong positive gradients that outline the shape of certain lines and curves at different rotations and angles. It is unclear why strong negative weights are in some of the corners. We speculate these could be to better inform the location network and it would be interesting to view the correlation of neuron activations and change in location values.

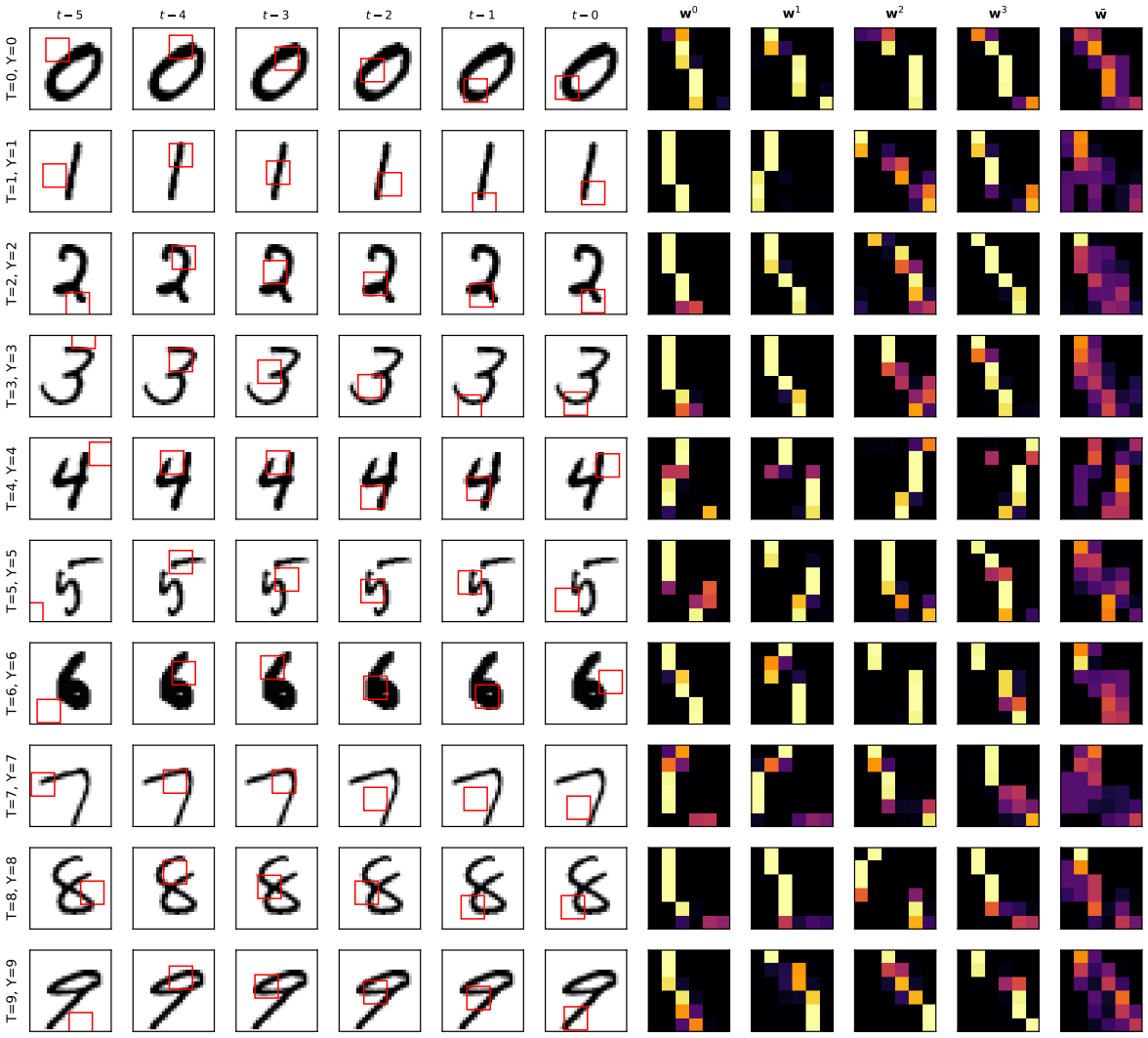



**Figure 3.6:** Example trajectories and distribution of attention weights for our best MNIST model.

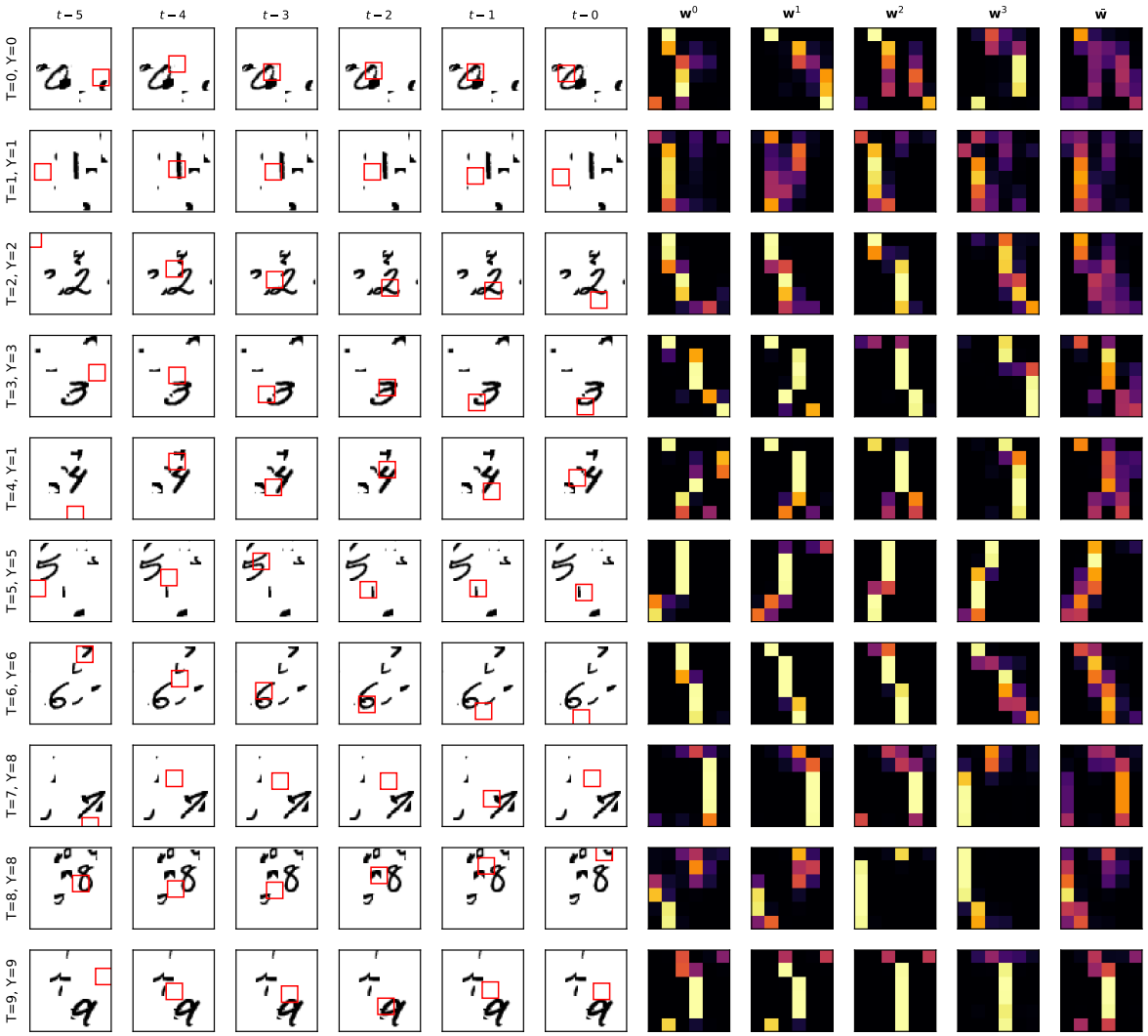



**Figure 3.7:** Example trajectories and distribution of attention weights for our best cluttered MNIST model.

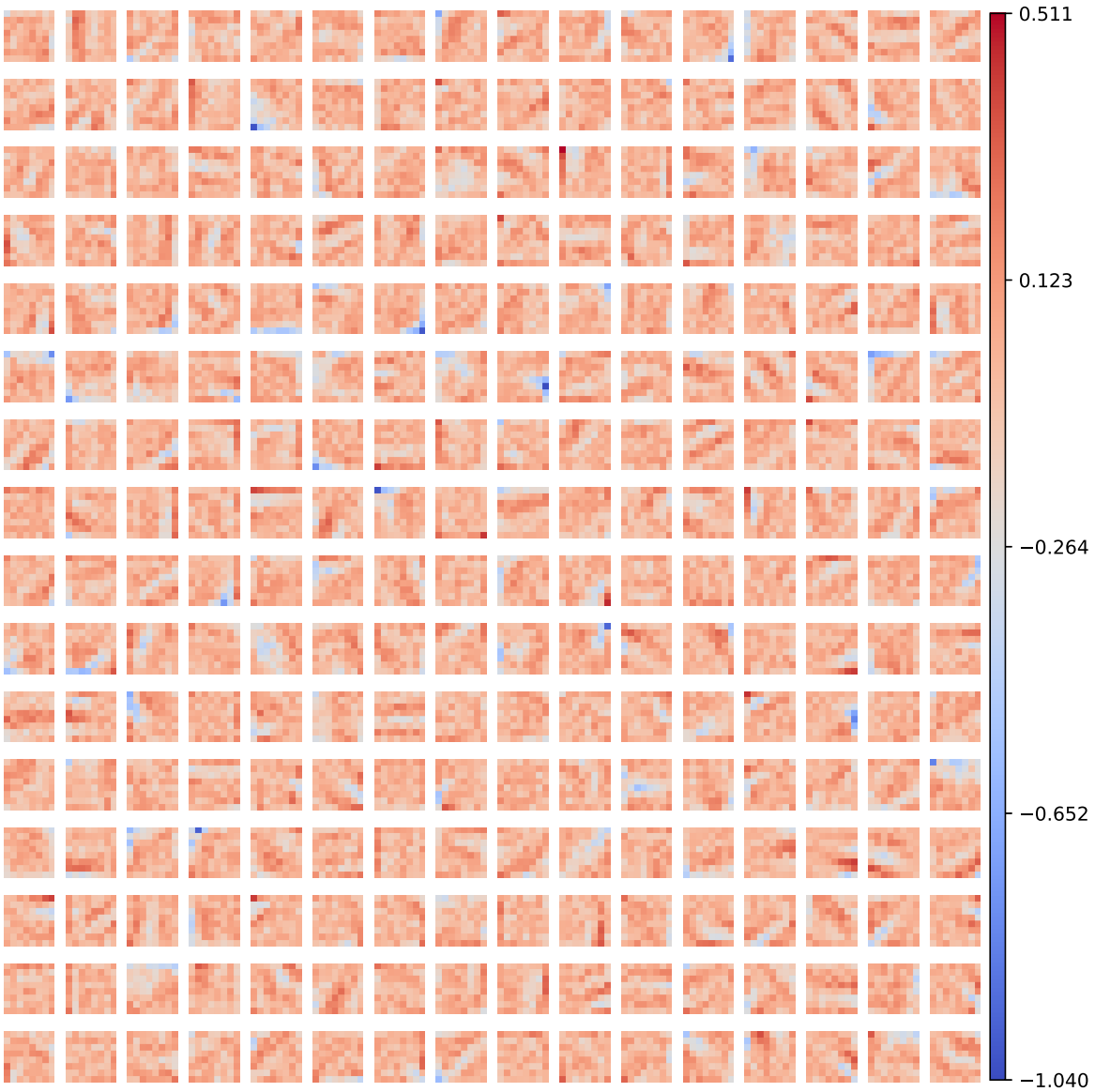



**Figure 3.8:** Input unit weights for the fully-connected layer in the glimpse network that takes as input an $8 \times 8$ glimpse of MNIST digits. Red indicates a higher, positive weight value and blue is smaller and negative.

## 3.4 Adaption to the Climate Domain

Detecting indicators of climate change within temperature and precipitation fields from annual variations is a challenging task. However, by extracting the forced signals amidst the background climate noise (from internal variability and model uncertainty), we can bring new insights to mitigation and adaptation science [100–103]. Following prior work, we train a neural network to predict the year from single-variable climate simulations. These works often use fully-connected and convolutional networks and perform a post-hoc analysis on the models, operating over the full input domain and learning potentially spurious correlations. In contrast, we show how our model of sequential attention (detailed previously in Section 3.2) can be adapted to distinctly highlight the most conspicuous regions, providing an interpretable model of decision-making.

There has been significant progress in using machine learning to make interpretations of climate simulations. As it relates to this work, [28] make interpretations by studying and visualizing a the regression weights from a single layer network. The work in [26] extends the analysis to nonlinear multi-layer networks using Layer-wise Relevance Propagation (LRP) [104] to identify and visualize the multivariate patterns learned by the networks. As motivated by [105], they use backward optimization to find an optimal input map for a given year to assess. The latter interpretations are then extended in [27] by using Alopex, a correlative learning and gradient free optimization method, to find optimal inputs over newer, similar datasets. Most recently, [25] trained networks with different input fields (varying the seasonality and input variables) from climate simulations, showing improved performance when multiple input fields are combined. Further, extending the approaches using LRP, they group relevancy maps with $k$-means clustering to identify distinct patterns.

The most interpretable models described above are linear or shallow networks, yet their performance is limited. On the contrary, the more complex models lack inherent interpretability and require the full input image, even if every location is not relevant and incur additional computation. Additionally, the backward optimization and explainability methods, while they do highlight relevant regions, emphasize even the extraneous regions, leaving additional unanswered questions of true relevance. Moreover, these studies often down sample the high-resolution maps, potentially losing explainable fidelity. Notably, however, the studies with interpretations from LRP are particularly motivating as they bring light to *some* localized regions with increased relevance (see Figure 8 and 9 in [26]). We hypothesize that sequential attention can identify these regions at inference time, without the need of additional explainability methods. Doing so would improve interpretability and encourage trust in our findings, given the model's prediction is constrained to smaller and direct locations on the globe.

In this section, we begin by introducing the dataset and baseline model in Section 3.4.1. Then we detail the relevant changes needed for our model of sequential attention to adapt to this problem, and generally for regression analysis, in Section 3.4.2. Lastly, in Section 3.4.3, we report on our results and discoveries found with sequential attention.

### 3.4.1 Preliminaries

Before we get to the changes needed for our model of sequential attention, we first introduce the dataset and describe the baseline model that is used to derive insights to our results.

#### Dataset Details

We use the annual-mean global $2\,\mathrm{m}$ air temperature $(\mathrm{K})$ output from climate simulations performed by the sixth phase of the Coupled Model Intercomparison Project (CMIP6, [106]). Although precipitation data is included in these simulations, our study focuses solely on temperature. CMIP6 provides $36$ model simulations from monthly runs for years 1850–2100, with a spatial resolution of $120$ latitude and $240$ longitude values (corresponding to each pixel). We compute the annual mean for each model, then subtract the global annual mean (across all models and years) to remove constant variability.

The target data are the individual years from which the means are computed. Each year has $36$ data samples from each simulation, totaling $9,036$ samples. Thus, our goal is to estimate the year given a climate map. Some prior work [25, 26] use the target as a label for fuzzy classification, while other studies [27, 28] take a regression approach. In this work, we adopt the latter framing, which requires some change to our architecture and learning algorithm (Section 3.4.2). Once preprocessed, data are partitioned into training $(7{,}279)$ and test $(1{,}757)$ sets as separated by model simulations.

In Figure 3.9, we show select samples from a single model corresponding to the given years. For improved visual clarity, each map has their data subtracted from the year 1850. This shows more clearly the relative changes, positive or negative, and the localized visual patterns for each year. A simple interpretation has relatively cooler temperatures over Russia in Figure 3.9a, while Figure 3.9c has an increase in temperature over the North and South Poles, relative to 1850. This illustration displays potential locations that could be indicators captured by sequential attention.

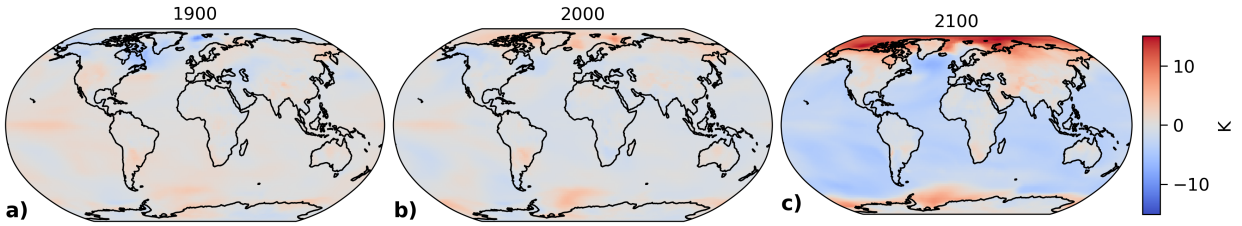



**Figure 3.9:** Global annual mean removed temperature with differences from year 1850 for visual clarity. Individual panels are from the same model but different years (column-wise) from CMIP6.

### Baseline Model Comparison

We reproduce results in [26], albeit with phase six data (CMIP6) and as a regression task as opposed to fuzzy classification, by training a network and computing the LRP maps. Using this result, and to provide greater support to our method, we compare the alignment of similar locations from these maps and the trajectories found with sequential attention.

The model is a two-layer fully-connected network with $10$ tanh hidden units that produce a single scalar output from the input maps. To stay consistent with [26, 28], we optimize the MSE with an $L_2$ norm penalty, i.e., ridge regression [107], calculated as

$$\mathcal{L}_\ell = \frac{1}{n}\sum_{i=1}^{n}(y_i - t_i)^2 + \frac{\lambda}{n}\|\mathbf{w}^{(1)}\|_2^2, \tag{3.9}$$

where $\mathbf{w}^{(1)}$ are network weights in the first hidden layer and $\lambda = 1\text{e}5$ is the ridge parameter influencing the update penalty. This penalty effectively reduces the multicollinearity, or the spatial correlation of the input across grid points, and produces generally smoother indicator maps on a local level. After experimenting with different $\lambda$ values, we found this value to produce smoother feature maps without sacrificing much detail. The network is trained for $500$ epochs (bs $= 251$) and is optimized with Adam ($\eta = 1\text{e}{-4}$).

## 3.4.2 Sequential Attention for Regression

The model of sequential attention introduced in Section 3.2 is outlined for classification tasks. The classification network emits a label that is used for a prediction, computing the cross entropy loss during training, and calculating the reward from correctly predicted targets for the policy gradient. For regression, we modify the three following components:

(a) *classification/output network* to predict a single continuous-valued scalar, $y_t = f_y(h_t; \theta_y)$, as opposed to a class label

(b) *loss function* to update all other network components with the mean-squared error (MSE), $\mathcal{L}_y = -\sum_{i=1}^{n}(t_i - y_i)^2$, instead of cross entropy

(c) *reward function* to be continuous, rather than discrete, as defined by the negative absolute difference between the target and output, $r = -|t - y|$

We propose additional architectural and data modifications to stabilize training and encourage exploration. To the former, we include batch normalization to the location, output, and baseline networks. The location network benefits greatly from this addition when estimating the mean before reparameterization. While the change is small, we find that exploration of more diverse locations are significantly encouraged early in training, improving overall performance. We attribute this to the networks having improved gradient flow and the hidden state, $h_t$, having a more consistent distribution. Additionally, in the case where the data distribution has a large variance, unlike that of MNIST (Section 3.3), we find it helpful to standardize the input and target data. This is done by z-score normalizing data to have zero mean and unit variance using the statistics of the training data.

We train our models on a single node with an NVIDIA RTX 3090 (24GB) GPU, Intel i9-11900F (2.50GHz), and 128GB memory. A hyperparameter sweep is done with 576 models, varying the network

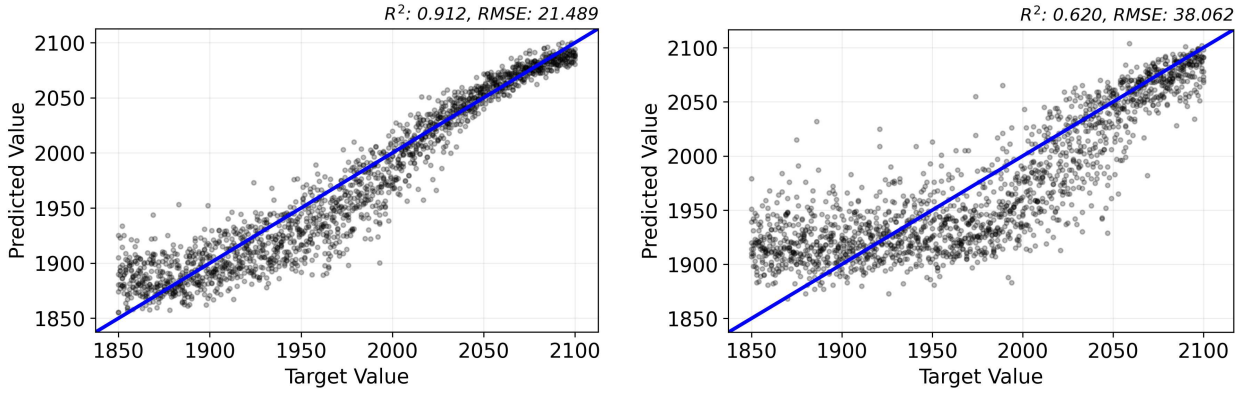
**Table 3.3:** Hyperparameters for CMIP6.

| HYPERPARAMETER | VALUE |
|---|---|
| EPOCHS (PATIENCE) | 150 (40) |
| LEARNING RATE $\eta$ | $3e-4$ |
| BATCH SIZE | 128 |
| GLIMPSE SIZE ($g \times g$) | 16 |
| NUMBER OF SCALES $s$ | 3 |
| SCALE MULTIPLIER $s_m$ | 2 |
| NUMBER OF GLIMPSES $K$ | 6 |
| GLIMPSE HIDDENS | 512 |
| LOCATION HIDDENS | 256 |
| LOCATION STD $\sigma$ | 0.05 |
| ATTENTION HEADS $H$ | 4 |
| ATTENTION HIDDENS | 256 |
| DROPOUT | 0.1 |

architecture and training values. Our best-performing model, given by the lowest validation RMSE, has parameters outlined in Table 3.3 and is used for further evaluations.

### 3.4.3 Experimental Results

Results are evaluated using the output (from the test data) of our best model and with the baseline model to draw insights from. We *quantitatively* assess performance with the root-mean-squared error (RMSE) and coefficient of determination ($R^2$). *Qualitatively*, we study the trajectories generated by sequential attention and compare them to the baseline model's attribution maps. The results aim to uncover the shifts to and correlations of regional patterns in climate models over time.




**(a)** Fully-connected network

**(b)** M-SAtt

**Figure 3.10:** Regression plot of target and predicted values with metrics in the title for each network.

Our model has an RMSE $= 38.062$ and $R^2 = 0.620$ compared to the baseline model with RMSE $= 21.489$ and $R^2 = 0.912$. While these values are worse performing, we observe various problem specific similarities and explainable insights. Figure 3.10 is a regression plot, showing how the target and predicted values align. A linear fit would have perfect skill. Although, instead, we observe more of a sigmoidal pattern in the predictions. The baseline's error (Figure 3.10a) reduces into the 21st Century and performs poorly prior to $\sim$1960. As we progress into later years, the increasing amplitude of forced changes makes it easier for the network to identify this period, despite the internal variability and internal climate model disagreement during these later years.

In Figure 3.10b, we observe a similar pattern to that of the baseline, but with a much larger spread and vertical shift at varying temporal periods. Between $\sim$1850-1960 there is an upward shift of $\sim$30 years,

suggesting that the learned locations are delayed compared to what is observed. For years ∼1960-2050 we observe the contrary, down by ∼30 years, where the predicted years are prior to those in the target data. Similar to the baseline, the best skill is captured in years after >2050.

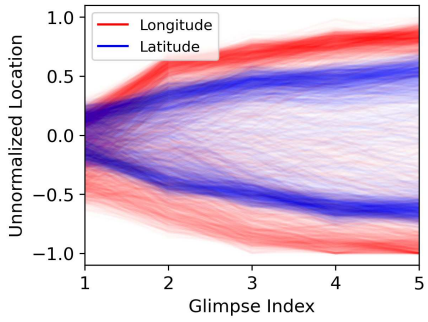



**Figure 3.11:** Unnormalized locations in $[-1, 1]$ for glimpse indices 1-5.

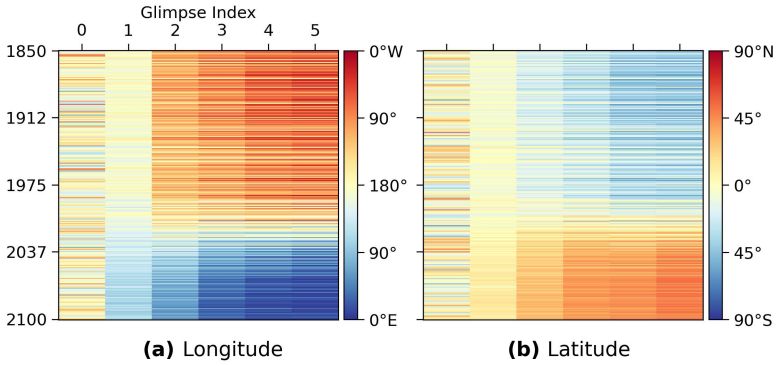



**Figure 3.12:** Predicted space-time locations for both **(a)** longitude and **(b)** latitude dimensions across all test samples.

Figures 3.11 and 3.12 show the predicted locations from our model for each glimpse as predictions made over all test samples. Recall the initial glimpse locations, $l_0$, is randomly sampled from a normal distribution. The unnormalized locations (Figure 3.11, a line per sample) shows the subsequent glimpse being sampled around zero, and quickly diverging primarily to two distinct regions. While some samples cover the space between, the regions are largely polarizing. Figure 3.12 has the location converted, coordinate space-time results visualized across samples for the longitude and latitude dimensions of $l_t$. This allows us to see when and where this divergence and shift in locations occur. Three predominate observations are seen in this figure.

Foremost, the first guided glimpse, $l_1$, is located around the equatorial dateline, and slightly biases northeast locations after year ∼2020. However, a more noticeable divergence occurs at this year, where earlier years are predicted from locations in the Southwestern Hemisphere ($\sim 90°\text{W}, 30°\text{S}$) and later predictions shift to the Northeastern Hemisphere ($\sim 120°\text{E}, 45°\text{N}$). Lastly, as additional glimpses are sampled, the model steadily hones in on these locations; it is as if the initial random glimpse is enough to guide the model in a particular direction. Incidentally, this shift around the globe at ∼2020 roughly corresponds to the improved performance observed in the regression plots in Figure 3.10b. This suggests that climate forcing present in later years are not only more accurately captured, but are done so nearest the North Pole and east of the Prime Meridian.

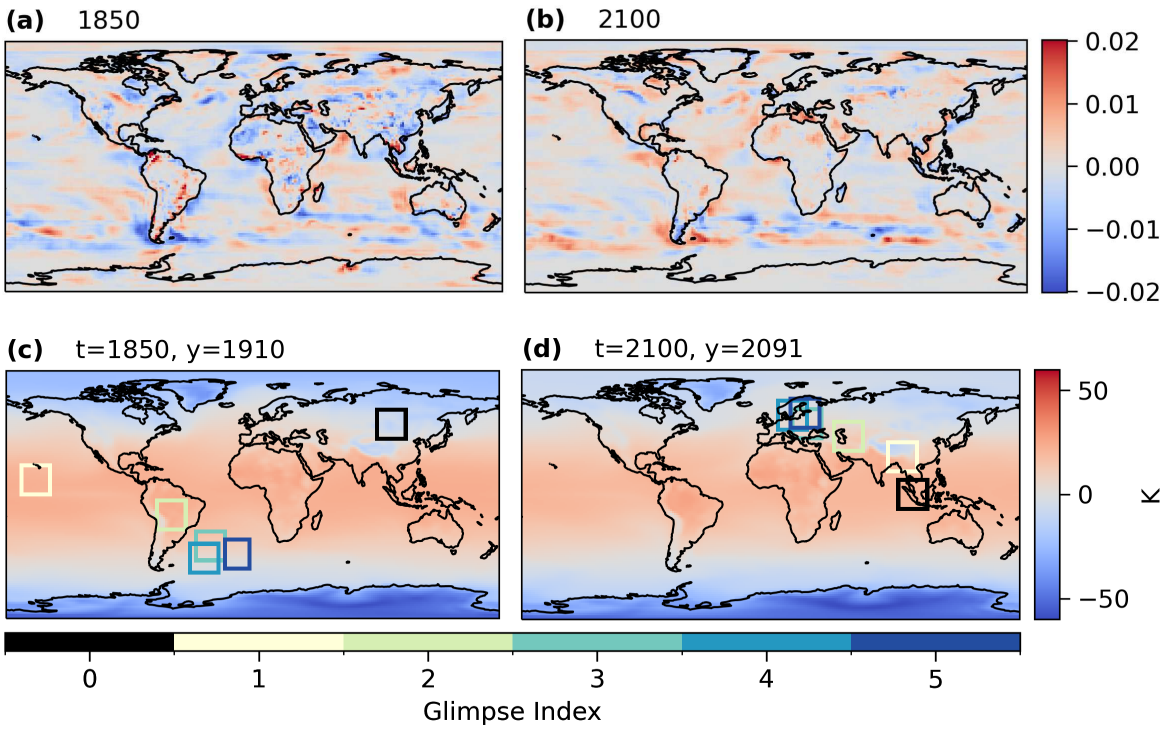



**Figure 3.13:** Sample comparison of attribution maps and locations found by sequential attention. The top row **(a,b)** is mean attribution, found with LRP on the fully-connected network, combined over all climate simulations corresponding to the years of 1850 and 2100. The bottom row **(c,d)** has the glimpse trajectories (indexed by $g_0$ in black with a gradient to $g_5$ in blue) from our model with corresponding target $t$ and predicted $y$ years.

Sample specific trajectories are made by superimposing the bounding box of glimpse locations over the input map. Recall that the information within these glimpses are all that is used to make a prediction. Two individual test samples are given in Figure 3.13c,d for target years 1850 and 2100, respectively. These locations align with divergence in Figure 3.12, which we further interpret using the attribution maps obtained by the baseline model. For each test sample, we compute LRP from the baseline and average the result across all simulations corresponding to a given year (Figure 3.13a,b).

From these relevancy maps, in 1850 we see the highest absolute relevance in South America, surrounding the southern oceans of Patagonia, scattered around Brazil, and in the northern region of Colombia. Moreover, there is constant higher absolute relevance covering other continents, such as Europe, Asia, and Africa. However, in 2100, these continents have significantly lower relevance in the baseline model. In fact, many regions with positive relevance is negated in these later years with the inverse in negative regions, but Europe, Asia, and Africa have a lower absolute relevance altogether. That is not to say all relevance is obsolete, rather

it is less prevalent. This change in relevance may be what our model is learning as part of where to look in later years.

## 3.5 Discussion

In this chapter we introduced a memory-based sequential attention model (M-SAtt) that combines information over a subset of image locations for both classification and regression tasks. Our proposed memory module incorporates a simplified transformer architecture at its core. This more directly allows for subsequent locations to be dynamically attended to and reweighed. By observing the attention weights corresponding to each location, we can directly explain the importance of these locations for the output prediction.

Our experimental results demonstrate that our model outperforms baseline architectures in traditional classification tasks. Although we do not achieve state-of-the-art performance, we consider this work a significant step toward enhancing model interpretability, aligning more closely with the principles of the biological visual system. This is especially pertinent for bridging the gap between machine learning and practical application domains, such as medical diagnoses and, in this work, modeling weather and climate.

As an extension to the classification setup, we demonstrate how M-SAtt can be adapted for regression tasks, specifically in identifying indicators of climate change. This adaptation involves modifying the output network to emit a scalar value, using mean-squared error as the loss function, and having a continuous reward function. Our experiments reveal that, while the model exhibits similar data biases to a simpler fully-connected network, it uncovers a notable spatial shift between the Southern and Northern hemispheres around the year 2020. This case study highlights the model's flexibility, showing how its components can be customized for various tasks and providing a foundation for future research on sequential attention in climate and weather modeling.

While our model does not achieve the best overall performance, the ability to explain which locations were used to form predictions is invaluable. Nonetheless, there are some additional limitations. Foremost, the choice of the initial glimpse location can slow down learning and potentially impact overall performance. Our experiments on smaller, classical vision tasks show less sensitivity to this, but the effects are more pronounced on large-scale imagery. A potential solution could involve using saliency measures to guide initial glimpse locations. Secondly, it remains difficult to determine whether the glimpse weights primarily influence the selection of subsequent locations or the final output prediction. Addressing this ambiguity through

gradient-based sensitivity measures would provide valuable insights and help disentangle the contributions of the glimpses to the model's decision-making process.

# Chapter 4

# Large-Scale Vision Transformers for High-Resolution Image Generation

In Chapter 3, we proposed a model of sequential attention with a transformer-based memory module that processes a subset of image patches, allowing us to dynamically focus on salient regions and incrementally build a prediction through a series of glimpses. While this approach proved effective for regression/classification tasks, it is not as practical for image-to-image translation, where we do want to process the entire image at once. Previous work has demonstrated the success of transformers in capturing complex spatial relationships within images, and we hypothesize that the use of contextual self-attention can be effective for image translation. Therefore, in this chapter, we shift our focus to the applied use of vision transformers for image-to-image translation in atmospheric science.

More specifically, in this chapter, we introduce a transformer-based neural network designed specifically to generate high-resolution ($3\,\mathrm{km}$) synthetic radar reflectivity fields at scale from geostationary satellite imagery (SRViT). Our aim is to enhance short-term convective-scale forecasts of high-impact weather events and to aid in data assimilation for numerical weather prediction across the United States. Our approach addresses the limitations of traditional convolutional methods, particularly their restricted receptive fields, by contextualizing over the full domain with transformers. The results demonstrate not only improved sharpness and accuracy across various composite reflectivity thresholds but also the potential for broader applications in atmospheric science. Additionally, we introduce a novel attribution method to guide domain experts in understanding model outputs, thereby improving the potential for explainability and supporting trustworthiness in interdisciplinary research.

We begin by motivating the importance and use of synthetic radar and discussing the limitations of existing methods in Section 4.1. Section 4.2 provides an overview of the dataset and baseline model used for comparison, followed by a detailed outline of our architecture in Section 4.3. We then discuss our attribution method in Section 4.4 and present the main findings, supported by case studies and ablation studies, in Section 4.5. Lastly, Section 4.6 summarizes our contributions and discusses the appropriate use of transformer models for meteorological applications.

Additional reading as it relates to this chapter can be found in the corresponding publication:

Stock, J., Hilburn, K., Ebert-Uphoff, I., & Anderson, C. (2024). *SRViT: Vision Transformers for Estimating Radar Reflectivity from Satellite Observations at Scale*. In ICML 2024 Workshop on Machine Learning for Earth System Modeling, July, 2024.

## 4.1 Background and Motivation

Accurate radar is crucial for operational forecasters to monitor and forecast the progression of high-impact weather events. Given the implications to public safety and agriculture, among others, this accuracy is key to protecting life and property. There are two primary uses of radar in weather forecasting: **(a)** enabling forecasters to view imagery and decide when to issue warnings, and **(b)** for imagery to be integrated into numerical weather prediction (NWP) models to forecast the weather [108, 109]. However, radar is limited to sparsely situated ground stations, leaving remote areas, mountains, and oceans with poor coverage. (Figure 4.2; see also Figure 1 in McGovern et al. [8]). To achieve more accurate forecasts, it is essential to improve radar coverage both spatially and temporally, which we address by using observational data.

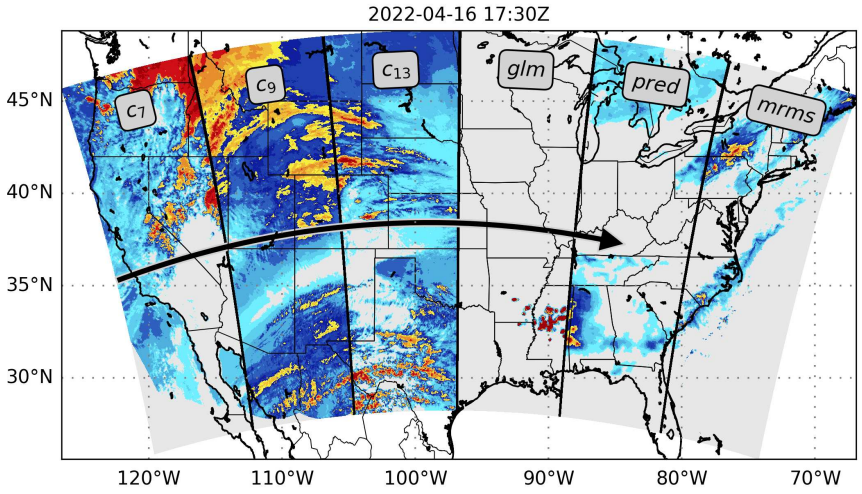



**Figure 4.1:** Full coverage input (columns $1-4$) and output prediction and ground truth (columns $5$ and $6$, respectively) example used to train SRViT. The real-time satellite observations enable forecasters to assess and forecast storm patterns at scale.

The Geostationary Operational Environmental Satellite (GOES) provides expansive coverage of the contiguous United States (CONUS) with $5\,\mathrm{min}$ updates and has been shown to be effective for operational weather forecasting [110]. With its low-latency and high spatial coverage, we believe there is a great opportunity in leveraging its geostationary imagers. In particular, our study focuses on utilizing imagery from

the Advanced Baseline Imager (ABI) [49] and the Geostationary Lightning Mapper (GLM) [62] onboard GOES-16 to generate high-resolution synthetic radar as modeled by the Multi-Radar Multi-Sensory (MRMS) product [111] over the CONUS domain. Figure 4.1 shows representative input and target output slices for a specific timestep to illustrate this goal.

Estimating radar reflectivity has recently received attention from the machine learning field. However, previous work has centered on incorporating physical indicators from numerical models [112, 113] or modeling smaller, localized spatial regions with convolutional models [114–116]. The transition to high-resolution, observation-based modeling remains in its early stages (Chapter 5 expands on this theme more generally). In this chapter, we address these issues by pairing large-scale observational data with a modified vision transformer [84] to generate high-resolution synthetic radar. The vision transformer captures long-range dependencies and attends to distant spatial features, overcoming the limited receptive fields of convolutional models, which is compelling for our task. To the best of our knowledge, this is the first study using a transformer for estimating radar reflectivity.

## 4.2 Preliminaries

Given this study centers primarily on a large-scale atmospheric science application, we begin by introducing the dataset in Section 4.2.1, detailing individual data features and composition, and in Section 4.2.2 we outline the baseline model used for making comparisons.

### 4.2.1 Dataset Details

This study uses observational data as *input* (ABI infrared Channels 7, 9, 13 and GLM) from the Geostationary Operational Environmental Satellite (GOES)-R Series satellites with *target* composite radar reflectivity from the Multi-Radar Multi-Sensor (MRMS) product—necessary variable details are provided below, with further details in [114]. We collect data in part from Hilburn [117, 118, 119] between 2018-2022 that span the contiguous United States (CONUS). As the intent of our results are to be used with data assimilation, we project the input and target samples to follow a $3\,\mathrm{km}$ High-Resolution Rapid Refresh (HRRR) mass grid, yielding $768 \times 1536$-pixel images ($2304 \times 4608\,\mathrm{km}$) on $6\,\mathrm{h}$ periods with a $15\,\mathrm{min}$ refresh ($96$ samples per day).

We restrict data to the warm season (i.e., April-September) where the strongest radar echos occur. Data are temporally partitioned to the years of 2018-2020 ($47{,}821$) for training, 2021 ($17{,}284$) for validation, and

2022 (17,344) for testing. Note the number of samples are indicated in parentheses that altogether total $1.9\,\mathrm{TB}$ in size. All samples undergo a preliminary quality control phase to ensure targets have the full range of radar reflectivity and there are no erroneous samples with large GLM and MRMS artifacts. Additionally, individual variables are scaled between $[0, 1]$ based on their corresponding histograms to stabilize training. When unstandardized, MRMS and model output values scale between $[0, 60]$ dBZ (unit of reflectivity).

**Advanced Baseline Imager (ABI)** We use Level-L1b radiances from GOES-16 ABI [49]. The imagery has a nominal spatial resolution of $2\,\mathrm{km}$, enabling the capture of fine-grained details, while ensuring a temporal refresh rate of $5\,\mathrm{min}$ for timely data updates. To account for both day-night scenarios, we concentrate on the use of the infrared bands, specifically: **Channel 7** ($3.7\,\mu\mathrm{m}$; Shortwave Window), **Channel 9** ($6.9\,\mu\mathrm{m}$; Upper-Level Water Vapor), and **Channel 13** ($10.3\,\mu\mathrm{m}$; Clean IR Longwave Window).

**Geostationary Lightning Mapper (GLM)** Accompanying the satellite imagery are real-time lightning observations from the GOES-R GLM [62, 120]. This instrument is a single-channel, near-infrared transient detector that monitors total lightning with a uniform spatial resolution of approximately $10\,\mathrm{km}$. Lightning itself is particularly useful for generating synthetic radar fields as their locations are often associated with strong updrafts within convective environments.

**Multi-Radar Multi-Sensor (MRMS)** We utilize quality-controlled composite reflectivity from the MRMS product [111] as our target dataset. This data combines information from different radar networks, surface observations, numerical weather prediction (NWP) models, and climatology. Through a sophisticated integration process, high-resolution mosaics are generated, offering precise spatiotemporal resolution. However, it is important to note that the coverage of the product is both limited in the vertical and spatial domain due to beam blockage and radar placement.

**Data Coverage and Distribution** Figure 4.2 illustrates the radar coverage from the training data, presenting a spatial map of cumulative echos. To represent the spatial distribution, we binarize the data by assigning a value of $1$ to composite reflectivity values greater than $0$, which are then summed across all samples. We then clip the result based on the $99$-th percentile to eliminate visual artifacts. Notably, the coverage is lowest over the western United States (in blue/gray), with a bias towards physical radar stations (circular red regions). The western region, affected by orographic enhancement, is further impeded by mountains (i.e., subject to

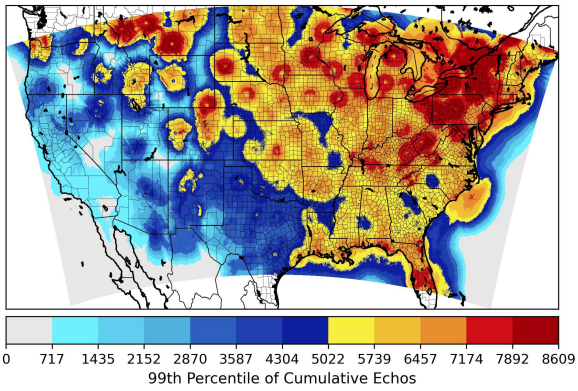



**Figure 4.2:** Spatial coverage of MRMS from the training data.

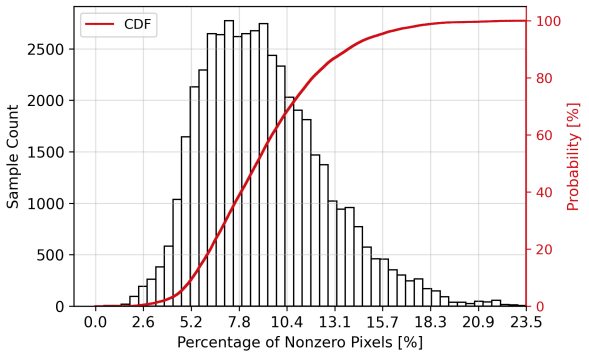



**Figure 4.3:** Sample histogram and cumulative density function of nonzero pixels present in the training data.

beam blockage), resulting in predominantly high-level atmospheric echoes that largely miss the stronger echoes near the surface. In the most extreme cases, precipitation may go completely undetected by radars in these regions. This further motivates the use of satellite observations to fill in these regions, but importantly, the lack of ground truth radar makes validation in these areas challenging.

In order to disentangle the sample specific coverage from the cumulative spatial representation, we compute the sample frequency of nonzero pixels with its cumulative density function (CDF). Figure 4.3 displays the result on the training data, showing a histogram with a slight positive skew and mean centered around $9.2\%$, indicating moderate coverage within samples relative to the CONUS domain. The CDF follows in that a substantial proportion of the training samples have nonzero pixel percentages above the median value. Importantly, this indicates that all samples have some level of coverage, irrespective of their spatial extent, without exceeding into high, erroneous values.

### 4.2.2 Baseline Model Comparison

The GOES Radar Estimation via Machine Learning to Inform NWP (GREMLIN) model [114] uses a fully-convolutional encoder/decoder architecture (without skip connections) that we denote as "UNet" in this study. Trained previously on smaller $256 \times 256$-pixel images and regionally filtered by storm reports (tornado, hail, and wind events), this architecture serves as the baseline for satellite to radar emulation. Using the same hyperparameters as in the original work, we retrain the UNet and both evaluate and compare its performance to our model using data from the entire CONUS domain—the first reproduction at this spatial scale.

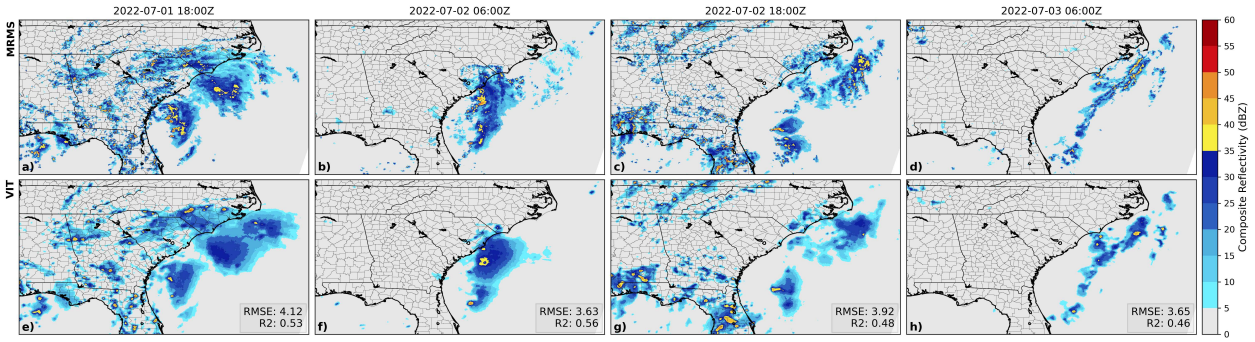



**Figure 4.4:** Temporal progression of Tropical Storm Colin as it compares MRMS (top row) and SRViT (bottom row). Sample RMSE and $R^2$ values are shown for each timestep (panel columns).

The encoder consists of three blocks with $3 \times 3$ convolutions followed by max-pooling and the ReLU activation function. The decoder begins by performing nearest neighbor upsampling before applying the convolution and activation, repeating this process until we achieve an equivalent spatial dimension to the input of the encoder. The output head is a single point-wise convolution with a $1 \times 1$ filter to combine the output feature maps into a single channel, representing MRMS. The encoder and decoder both use convolutions with 32 filters, resulting in a network with $47{,}457$ trainable parameters.

## 4.3 Satellite-to-Radar Vision Transformer (SRViT)

Our proposed model extends the vision transformer [84] and is designed specifically for image-to-image translation. We include the training setup with hyperparameters and hardware specifications in Section 4.3.1.

Consider an image $I \in \mathbb{R}^{c \times h \times w}$ that is partitioned into equally divisible patches (synonymous with tokens) of size $p$ as $x = \{x_0, \ldots, x_n\}$, where $x_i \in \mathbb{R}^{d_{in}}$ and $n = (w/p) \cdot (h/p)$ with $d_{in} = p^2 c$. Once partitioned, the set of tokens $x$ are linearly projected to dimension $d < d_{in}$ and we add a standard sine-cosine positional encoding. Let matrix $\mathbf{X}^l \in \mathbb{R}^{n \times d}$ be the new row-wise concatenation of the tokens. A typical transformer block $\phi$ at layer $l$ processes the set of tokens with multi-head self attention (MSA) and a point-wise fully-connected network (FCN) as,

$$\phi(\mathbf{X}) = \text{FCN}(\text{MSA}(\mathbf{X})) \quad \text{such that} \tag{4.1}$$

$$\text{MSA}(\mathbf{X}) = [\mathbf{O}_1, \mathbf{O}_2, \ldots, \mathbf{O}_h]\mathbf{W}^{\mathbf{O}}, \tag{4.2}$$

where $h$ is the number of heads, $\mathbf{W}^{\mathbf{O}} \in \mathbb{R}^{hv \times d}$ are trainable weights, $[\cdot]$ is the column-wise concatenation, and $\mathbf{O}_i \in \mathbb{R}^{n \times v}$ is the output of the $i$-th attention head with latent dimension $v < d$. We compute each head

as,

$$\mathbf{O}_i = \mathbf{A}_i \mathbf{V}_i \quad \text{such that} \tag{4.3}$$

$$\mathbf{A}_i = \text{softmax}\big(\mathbf{Q}_i \mathbf{K}_i^\top / \sqrt{d}\big) \in \mathbb{R}^{n \times n}. \tag{4.4}$$

The queries, $\mathbf{Q}_i$, keys, $\mathbf{K}_i$, and values, $\mathbf{V}_i$ are found via a linear projection of $\mathbf{X}$ by,

$$\mathbf{Q}_i = \mathbf{X}\mathbf{W}_i^Q, \quad \mathbf{K}_i = \mathbf{X}\mathbf{W}_i^K, \quad \mathbf{V}_i = \mathbf{X}\mathbf{W}_i^V, \tag{4.5}$$

with trainable weight matrices $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V \in \mathbb{R}^{d \times v}$. The FCN block at layer $l$ is a two layer network separated by the Gaussian Error Linear Unit (GELU) activation, $\delta$, and dropout (with a default $p = 0.2$) that takes as input the layer normalized output of the MSA, $\bar{\mathbf{X}} = \text{LN}(\mathbf{X})$. This block is computed as,

$$\text{FCN}(\bar{\mathbf{X}}) = \delta(\bar{\mathbf{X}}\mathbf{W}^R)\mathbf{W}^S, \tag{4.6}$$

where $\mathbf{W}^R \in \mathbb{R}^{d \times m}$ and $\mathbf{W}^S \in \mathbb{R}^{m \times d}$ such that $m > d$.

Each transformer block has independent weight matrices and yields a new set of tokens of the same dimension given by $\phi : \mathbf{X}^l \to \mathbf{X}^{l+1}$. After $L$ transformations, we introduce a linear decoding block and reshaping operation $f$ to reconstruct the intermediate output. This is denoted by,

$$z_0 = f(\mathbf{X}^L) = \text{reshape}(\mathbf{X}^L \mathbf{W}^F, (c \times h \times w)), \tag{4.7}$$

where $\mathbf{W}^F \in \mathbb{R}^{d \times d_{in}}$ and $z_0$ is the same dimension as the input image $I$. The output of this layer is a close approximation to the output, but as shown in Section 4.5.4, we find it to be inaccurate. Therefore, we include convolutional layers following $f$ to effectively smooth the boundaries of the decoded tokens via weight sharing. We represent the $N$ subsequent convolutional layers as,

$$z_i = \text{ReLU}(\mathbf{W}_i^C * z_{i-1}), \text{ for } i = 1 \text{ to } N, \tag{4.8}$$

where $*$ denotes the convolution operation, $\mathbf{W}_i^C$ are the trainable weights at layer $i$, and ReLU is applied element-wise. The model's final output, $y$, when $i = N$ has the same spatial dimension as $I$ with a single output channel.

### 4.3.1 Training Details

We use the weighted loss from Hilburn et al. [114] to balance the rare but high radar reflectivity values with the small but common values. Using the observation that the probability density function of the scaled target samples are approximated by $P(t) \propto e^{-5t}$, an exponential factor is introduced as. Specifically, denoted as

$$\mathcal{L}_e = \frac{1}{m} \sum_{i=1}^{m} \exp\left(w_0 t_i^{w_1}\right) \cdot (y_i - t_i)^2, \tag{4.9}$$

where $t$ and $y$ are the ground truth and predicted values, respectively, $m$ is the number of pixels, and $w_0 = 5$ and $w_1 = 4$ are found by optimizing the categorical bias of model performance.

We develop our models in PyTorch and train them on a cluster of $8\times$ $40\,\mathrm{GB}$ NVIDIA A100 GPUs for a maximum of $300$ epochs, albeit with early stopping, using AdamW and saving the best-performing model evaluated on the validation data during training. End-to-end training of our model takes about $100\,\mathrm{h}$ wall-clock time, and due to high compute and limited accessibility, we perform minimal hyperparameter tuning (see Table 4.1 for final values) and report results of a single trial; our final network consists of $643{,}105$ trainable parameters.

## 4.4 Attention via Token (Re)Distribution

The activation of scaled dot-product attention computes the matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ that represents the reweighing of our values from the combined queries and keys (see Equations (4.3) and (4.4)). Some prior work aims to interpret this matrix as importance or relevancy scores [121–123]. Indeed, these weights capture the interactions between all tokens, but faithful interpretations are challenged by not considering **(a)** the token vector magnitudes and high dimensionality, and **(b)** the interplay between tokens across intermediate hidden layers [124–126].

To address these issues, we introduce Token (Re)Distribution, a transformer-based attribution method that captures the interaction of all other tokens, as learnt by the network, for a particular token. This approach

allows for interpretations of how the value of input tokens are redistributed amongst the others. Our method is based on prior work [127–129] that use the gradient of intermediate layers and self-attention features. However, the key difference is in how we summarize gradient information across the $d$-dimensional token embeddings to highlight the relevance of individual tokens, which is particularly useful for the wide class of transformer-based models.

### 4.4.1 Technical Details

Recall that transformer blocks are characterized by the function $\phi : \mathbf{X}^l \rightarrow \mathbf{X}^{l+1} \in \mathbb{R}^{n \times d}$ for $l = 0 \ldots L-1$ layers, where $n$ is the number of tokens and $d$ is the embedding dimension. Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be the input token embedding (prior to positional encoding) and $\mathbf{Z} \in \mathbb{R}^{n \times d}$ be the output of a given block.

Our goal is to compute a sensitivity matrix $\mathbf{U} \in \mathbb{R}^{n \times n}$, where each element $\mathbf{U}_{ij}$ represents the influence of the $j$-th input token on the $i$-th intermediate token in layer $l$. To achieve this, we consider the Jacobian $\mathbf{J} \in \mathbb{R}^{n \times d \times n \times d}$, where $\mathbf{J}_{ijkl} = \partial \mathbf{Z}_{ik} / \partial \mathbf{X}_{jl}$, although in PyTorch, we can more efficiently compute vector-Jacobian products. We do this with an all-ones vector $\mathbf{1} \in \mathbb{R}^d$ for each $d$-dimensional intermediate token $\mathbf{z}_1, \ldots, \mathbf{z}_n \in \mathbf{Z}$ as

$$\mathbf{g}_i = \mathbf{1} \cdot \frac{\partial \mathbf{z}_i}{\partial \mathbf{X}} = \sum_{k=1}^{d} \frac{\partial (\mathbf{z}_i)_k}{\partial \mathbf{X}} \in \mathbb{R}^{n \times d}. \tag{4.10}$$

This provides a measure of the total sensitivity of the token to changes in the input. However, the sensitive to the embedding dimensions of the input is not as informative, so we define a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ to reduce the gradients to a scalar. There are many operators for $f$, including the L$p$ norm, but we find the absolute value of the sum of gradients be the most salient, specifically,

$$f(\mathbf{g}_i)_j = \left| \sum\nolimits_{k=1}^{d} (\mathbf{g}_i)_{jk} \right|. \tag{4.11}$$

Thereafter, we construct the matrix $\mathbf{U}$ by iterating over each intermediate token in layer $l$ as

$$\mathbf{U} = \left[ f(\mathbf{g}_1), f(\mathbf{g}_2), \ldots, f(\mathbf{g}_n) \right]^\top. \tag{4.12}$$

This provides a holistic view for how the change in the input token, over its entire embedding, affects the intermediate token; a large magnitude is highly responsive to small changes. Thus, we define a redistribution of those input tokens, as a result of self-attention, to the value of an intermediate token, i.e., Token

(Re)Distribution. With multiple transformer blocks (or layers), the mean given by $\bar{\mathbf{U}} = \frac{1}{L}\sum_{l=1}^{L}\mathbf{U}^{(l)}$ can be more informative, and allow for for visualizations of individual tokens using data indexed at a given row.

Forthcoming are specific examples and case studies where we will discuss the interpretations of Token (Re)Distribution for our application in Section 4.5.3.

**> vignette (2):** ***intuition from traditional benchmarks***

To demonstrate and evaluate Token (Re)Distribution, we study the pretrained Data-Efficient Image Transformer (DeiT) [130]—a commonly used baseline model for many transformer studies. Specifically, we use DeiT-T, a $12$ layer, $5$M parameter model with a patch size $p = 16$ that was trained on ImageNet [131], achieving $72.2\%$ top-1 accuracy. A naive attempt at interpreting the distribution of attention can be done by visualizing the attention matrix. Figure 4.5 illustrates this on an arbitrary data sample for a single head. Evidently, as the number of layers and heads increase, the explanatory capacity of these weights become non-trivial.

As a result, we turn to Token (Re)Distribution to glean more insights. Specifically, we compute our method over individual correctly classified data samples and qualitatively evaluate performance by focusing on a select number of tokens within these samples and discuss the result. All experiments are done in PyTorch using automatic differentiation for gradient computations.

We begin by precomputing Token (Re)Distribution across all layers for a single sample. Figure 4.6 shows this result with the general distribution of tokens. While these observations are unique to this sample, we detail guidelines for assessing the result. We find for the first eight layers, the intermediate tokens $\mathbf{Z}$, generally have high magnitude contributions from positionally close input tokens. Further, we see a horizontal and vertical hatching among groups of input tokens for these layers. The final four layers emphasize a group of tokens with those nearest the center having the highest weight. It is not clear if these patterns are an artifact of the model or specific to the data sample, and an additional study of global pattern analysis would be beneficial.

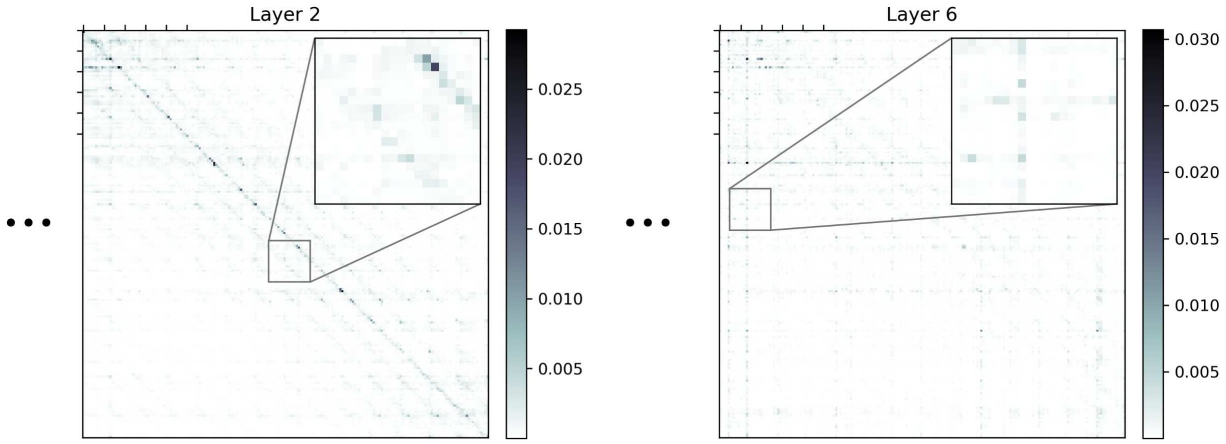



**Figure 4.5:** Magnitude of the attention weights from DeiT-T on ImageNet for a single head in different layers from a forward pass of an arbitrary data sample. Interpretations of these weights are non-trivial.

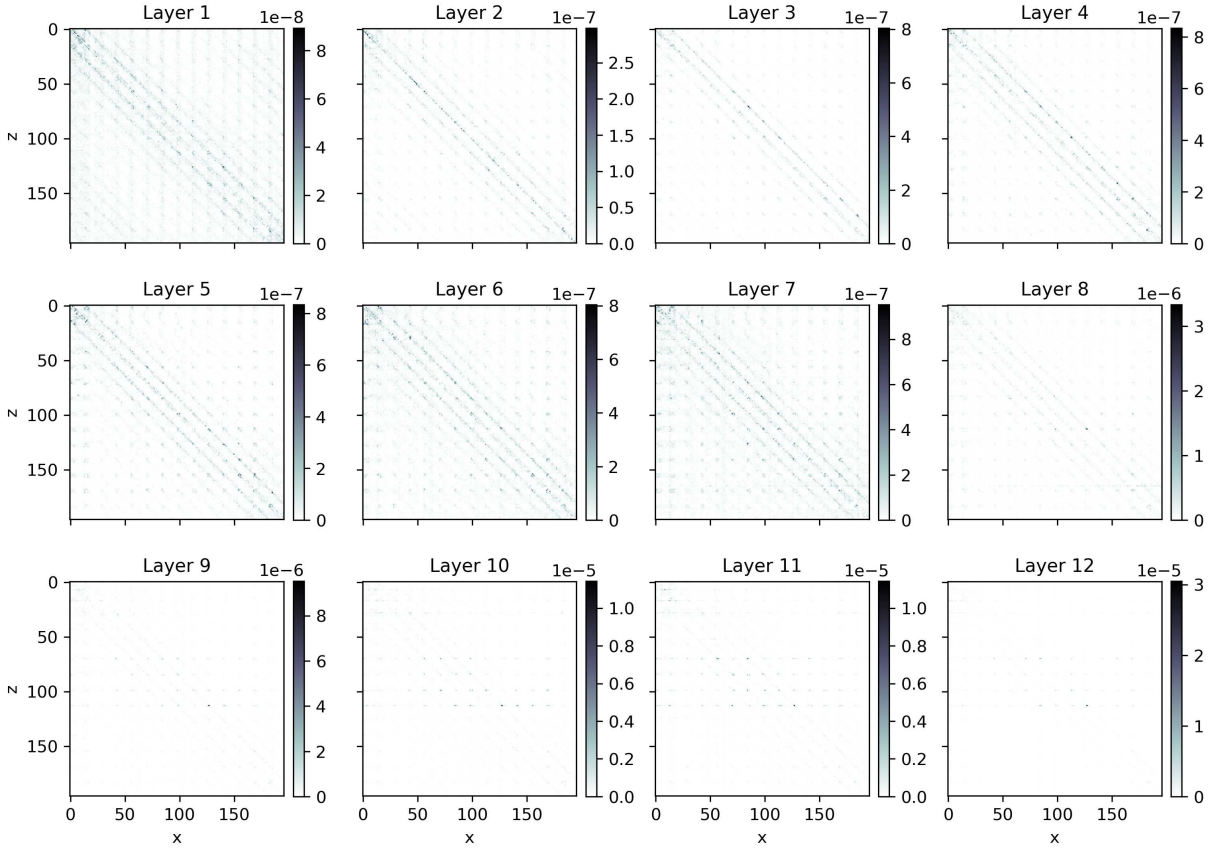



**Figure 4.6:** Layer-wise Token (Re)Distribution for a sample image (the owl in the top row of Figure 4.7) with the intermediate tokens, $z$, on the y-axis and the input tokens, $x$, on the x-axis.

These heatmaps can be difficult to interpret without context of the original input image. As such, we take the mean over all layers and superimpose the normalized magnitude on the image. A visualization

of this mean, for a subset of samples, indexed at different rows in $\mathbf{U}$ are shown in Figure 4.7. For each sample, we highlight in yellow the token of interest chosen at different locations throughout the image.

Interestingly, we see for the top row, the auxiliary tokens surrounding the owl (a,d) are most sensitive to similar tokens along the periphery. These tokens have similar qualities, e.g., branches, foliage, bokeh, etc., with a distinct boundary around the owl. The central tokens, focusing on the eye (b,c), are most sensitive to the opposing eye and the owl's beak. In the middle row, the trees to the left of the barn (e) are most similar and influential in the token's value. Those focused on the barn (f,g) are sensitive to neighboring barn tokens, where in both cases, the grass below has no influence. The token in the grass (h), however, is highly sensitive to neighboring tokens that are also in the grass. In the last row is an orchid with different tokens on the flower and background selected. We find the tokens on the pedals (i) to be a redistribution of the pedals, even those on the opposing side of the flower. In contrast, the background token (l) very distinctly accentuates the background with a distinct outline of the tokens surrounding the plant.

These findings do not explicitly detail the tokens that are used for classification. Instead, they illustrate the spatial flow of information within the image, bringing insight to how the value of an intermediate token is influenced by all of those in the input. This is useful to better understand the spatial relationships between tokens and the information that they are comprised of.

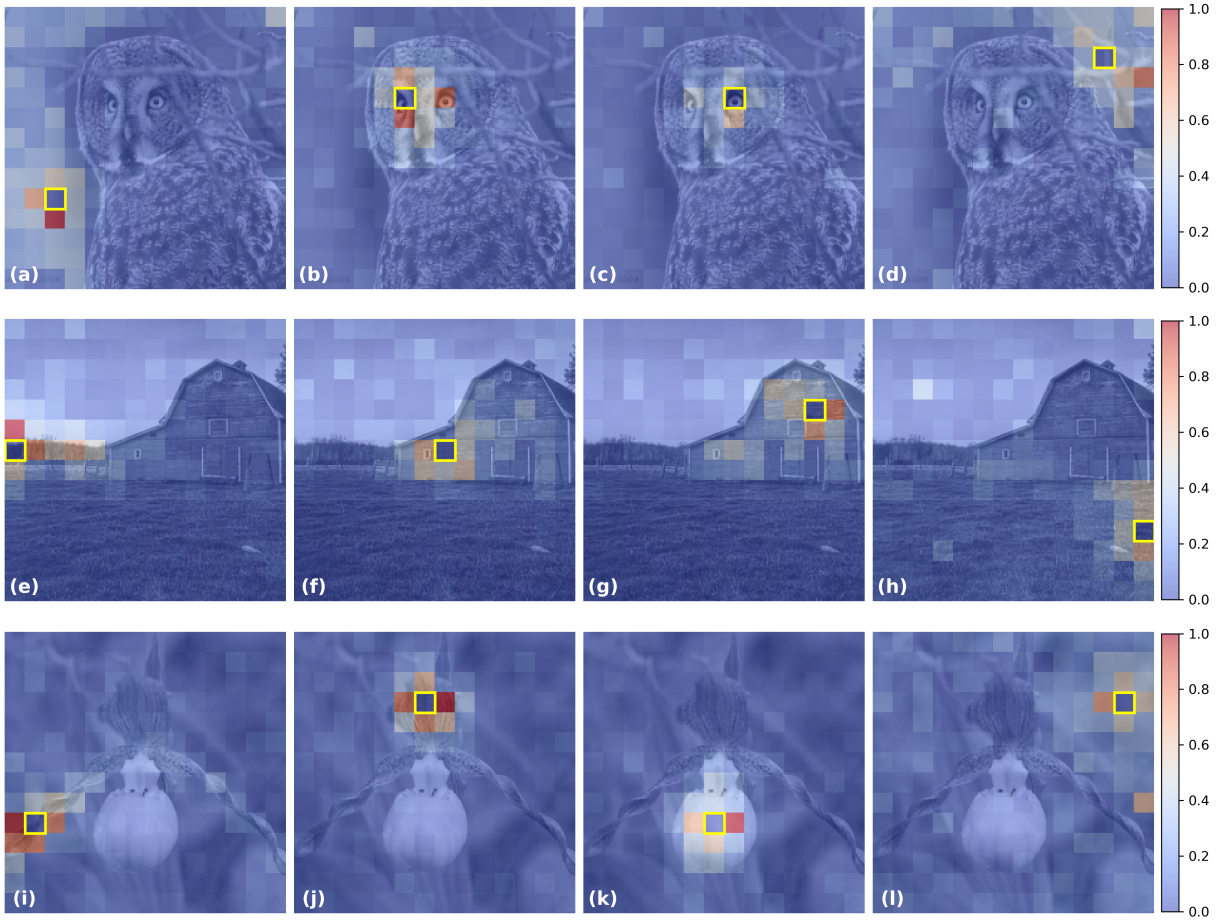



**Figure 4.7:** Sample min-max normalized gradient magnitude of Token (Re)Distribution for the (yellow; separate in each panel) token of interest overlaid on correctly classified samples (rows: owl, barn, and orchid).

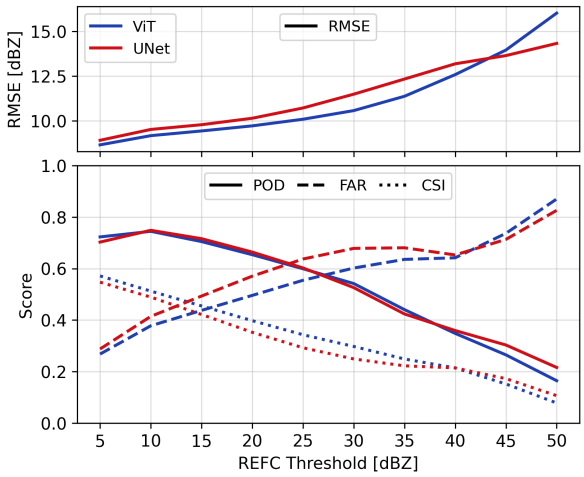



**Figure 4.8:** Categorical metrics at varying composite reflectivity thresholds as compared to the baseline.

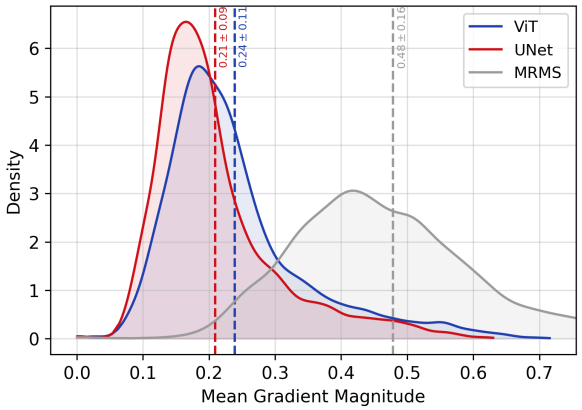



**Figure 4.9:** Kernel density estimation (KDE) of mean gradient magnitude of composite reflectivity over all test samples. The dashed line represents the mean with standard deviation.

## 4.5 Experimental Results

Primary results and comparisons with the baseline model are discussed in Section 4.5.1, and in Section 4.5.2 we investigate individual case studies of severe weather events, e.g., Tropical Storm Colin that we visualize in Figure 4.4.

### 4.5.1 Main Findings

We evaluate our results with three unique categories of methods, including: **(a)** *standard metrics*: root-mean-squared error (RMSE) and coefficient of determination ($R^2$); **(b)** *categorical metrics*: probability of detection (POD), false alarm ratio (FAR), and critical success index (CSI) at different composite reflectivity thresholds; and **(c)** *sharpness quantification* with the mean magnitude of image gradients. We compute these metrics over the test data and make comparisons with the baseline UNet and ground truth MRMS.

**Model Performance** SRViT shows superior performance with the standard metrics, having an RMSE $=$ $3.09$ dBZ and $R^2 = 0.572$, outperforming the UNet with RMSE $= 3.21$ dBZ and $R^2 = 0.488$ (Table 4.2). These statistics capture general pixel-wise improvement, but are biased toward low coverage and zero-valued pixels. To improve our intuition of model performance, we evaluate the results on a linear interval where composite reflectivity is greater than predefined threshold levels. This more accurately displays where the improvements are most noticeable, i.e., which reflectivity levels are more accurately represented.

Most notably, in Figure 4.8, we find SRViT to exhibit the best performance at low- to mid-value thresholds. Relative to the UNet, the greatest improvements represented by FAR, CSI, and RMSE are between $[5, 40)$ dBZ. Specifically, there is a reduction of $0.96$ dBZ in RMSE for reflectivity greater than $35$ dBZ, while

**Table 4.1:** SRViT hyperparameters.

| Hyperparameter | Value |
|---|---|
| Global Batch Size | 16 |
| Learning Rate $\eta$ | $1e-4$ |
| Epochs (patience) | 300 (50) |
| Patch Size ($p \times p$) | $(12 \times 12)$ |
| Model Depth $L$ | 6 |
| Heads (per block) | 12 |
| Model Dimension $d$ | 256 |
| Inner Dimension $v$ | 64 |
| FCN Dimension $m$ | 512 |
| Conv Hiddens (filters) | $[32, 16]$ |

**Table 4.2:** Metric summary across models.

| Model | $\downarrow$ RMSE | $\uparrow R^2$ | $\uparrow$ Sharpness ($g$) |
|---|---|---|---|
| MRMS | – | – | $0.48 \pm 0.16$ |
| UNet | 3.21 | 0.488 | $0.21 \pm 0.09$ |
| Base-ViT | 3.05 | 0.487 | $0.21 \pm 0.09$ |
| SRViT | 3.09 | 0.572 | $0.24 \pm 0.11$ |

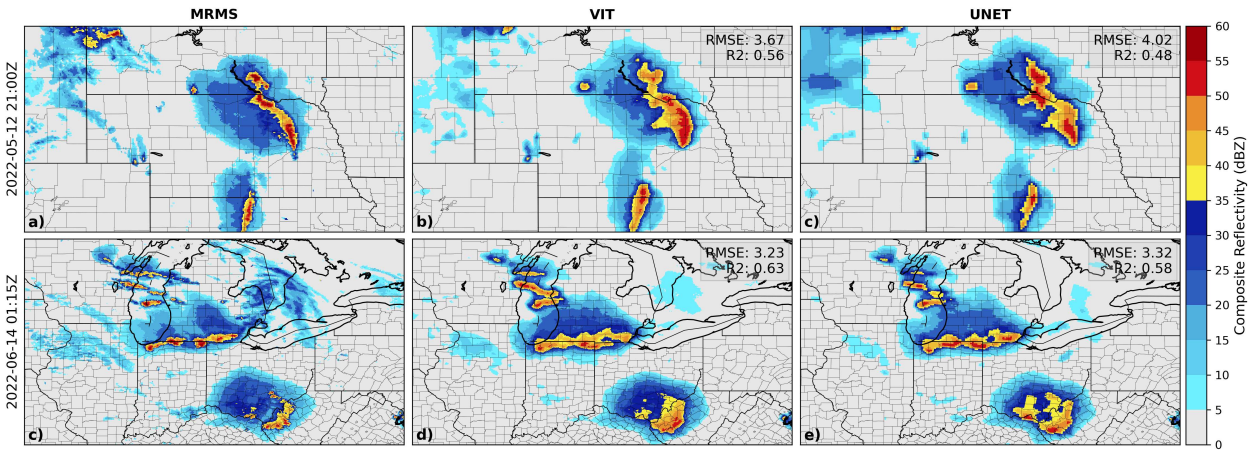



**Figure 4.10:** Northern Plains Derecho in panels (a-c) and Midwest Squall Lines in panels (d-f). Sample RMSE and $R^2$ values are shown for each case between model output (panel columns two and three) and MRMS.

FAR increases by $0.08$ at a threshold of $25$ dBZ. It is important to highlight that the POD of each network is consistent across thresholds levels until a divergence occurs at $40$ dBZ. At these higher thresholds ($> 40$ dBZ), the UNet displays marginally better performance across all the categorical indicators, which may be attributed to specific test cases.

**Evaluating Sharpness** A shortcoming of many neural network approaches to image-to-image translation, in particular those comprised of convolutions and pixel-wise, mean objective functions, are their abilities to represent sharp edges in the output. This effect yields outputs that are qualitatively fuzzy, an observation we find on the boundaries of composite reflectivity from the UNet. To quantify this observation, we compute the mean of the gradient magnitude for each output and compare the distributions amongst individual models. Specifically, we take the spatial mean after convolving a Sobel filter, i.e, $g = \frac{1}{m}\sum_{i=1}^{m}(G_{x_i}^2 + G_{y_i}^2)^{\frac{1}{2}}$, where $x$ and $y$ are image directions.

Figure 4.9 shows the kernel density estimation (KDE) plots of $g$ across all test samples for each model. Both neural network approaches exhibit positively skewed distributions with lower mean values compared to MRMS. With a direct comparison using Welch's independent samples t-test, we find that SRViT demonstrates a statistically significant improvement in sharpness over the UNet (t-statistic $= 27.71$, p-value $< 0.001$). SRViT has a mean gradient magnitude of $0.24 \pm 0.11$, whereas the UNet has a mean of $0.21 \pm 0.09$. These finding coincide with the qualitative assessments of observations (e.g., Figure 4.4) and suggest that our method enhances the sharpness of composite reflectivity. However, it is worth noting that both approaches still fall short of fully capturing the sharpness within the ground truth MRMS data.

### 4.5.2 Case Studies

Figure 4.4 illustrates the progression of Tropical Storm Colin on $12\,\mathrm{h}$ intervals from the output of SRViT on an Equidistant Cylindrical map projection. The short-lived storm had a peak intensity of $35$ knots from 2330 UTC July 1, 2023 to 1200 UTC July 2, 2023 over the western Atlantic. Accompanied with deep convection, Colin brought at most $19.28$ cm of rainfall to Wadmalaw Island, South Carolina [132]. SRViT adequately captures Colin's structure (Figure 4.4e) while accurately representing inland scattered convection (Figure 4.4g,h). This observation demonstrates its ability to handle complex weather patterns over the entire CONUS domain.

By enlarging model output, we study the results of our approach over a small spatial region for two other severe weather events and compare the predictions with MRMS and the UNet. In Figure 4.10, the top row shows a Northern Plains Derecho and the bottom row shows Midwest Squall Lines. In both cases, SRViT has an improvement to the standard metrics (lower RMSE, higher $R^2$) relative to the UNet. Furthermore, qualitative assessments have composite reflectivity boundaries with precise transitions, validating our findings of generating sharper output. The UNet also appears to overestimate for higher thresholds, whereas SRViT more accurately captures low- to mid-level thresholds but underestimates composite reflectivity $> 50$ dBZ. For example, within the lower squall line over Southern Ohio in the bottom row of Figure 4.10. These observations are consistent with the quantitative metrics we report in Section 4.5.1.

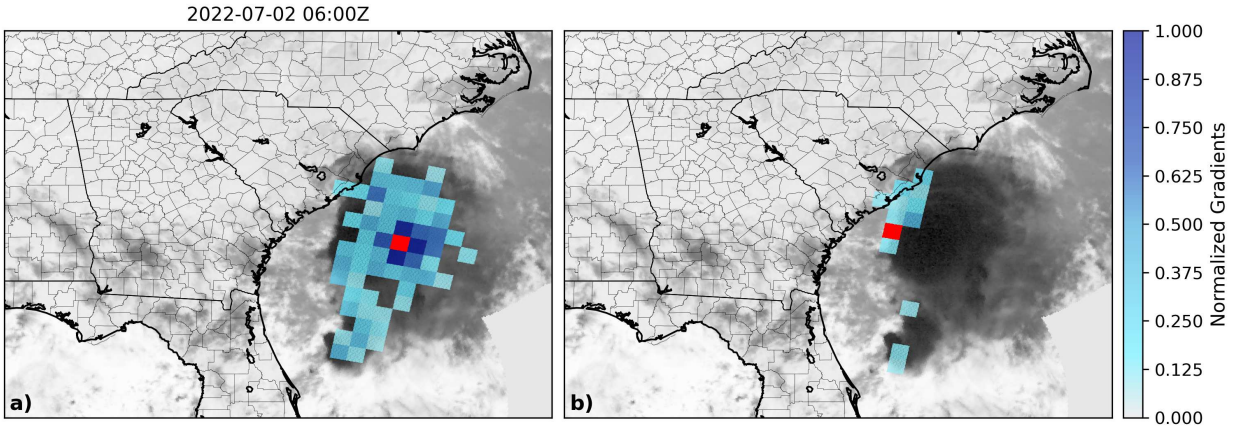



**Figure 4.11:** Sample min-max normalized gradient magnitude of Token (Re)Distribution for the (red; separate in panels **(a)** and **(b)**) token of interest overlaid on GOES-16 ABI Channel 7 (black indicates higher radiance).

### 4.5.3 Explanations from Token (Re)Distribution

In Section 4.4, we introduced an attribution method to uncover the contributions that input tokens have on their latent representations, termed Token (Re)Distribution. While the prior experiments were conducted on a pre-trained transformer for classification, we show how our method can as easily apply to SRViT. In fact, the calculations are merely identical to what is described previously, but the interpretations are even better suited for image-to-image tasks. As the intermediate tokens are linearly projected into the output domain (after the linear decoding block and before the convolutions), we can precisely determine how individual tokens in the input influence SRViT's output.

Leveraging Equations (4.11) and (4.12), we generate an attribution map for a sample of Tropic Storm Colin overlaid on Channel 7 of GOES-16 ABI in Figure 4.11. Each panel shows the result of a different token, from the same sample, with the normalized gradient magnitude on our common map projection. In Figure 4.11a the token of interest (in red), centered on higher radiance values (more opaque, in black) shows surrounding tokens within the clouds contributing to its value. Similarly, in Figure 4.11b, the token on the storm edge uses the values of others along this edge. Both cases are meteorological supported as it relates to the storm structure and extent of precipitation. Altogether, this method is a step toward guiding domain experts' interpretations of model output.

### 4.5.4 Architectural Ablation Study

To better understand our network design, we ablate the key components and discuss the qualitative and quantitative implications. Following the transformer blocks in SRViT are convolutional layers that effectively smooth the boundaries of decoded tokens. By training the base transformer (Base-ViT) model with the same hyperparameters and training procedure but without convolutions, we can study this smoothing observation.

We find quantitative statistics of Base-ViT having an RMSE $= 3.05$ dBZ and $R^2 = 0.487$. This shows a lower pixel-wise error to SRViT, which means its predictions are, on average, closer to MRMS. However, the lower $R^2$ suggests that the use of convolutions better explains a larger proportion of the variance and captures the underlying patterns more effectively. When assessing sharpness, it becomes evident that Base-ViT, having a mean gradient magnitude of $0.21 \pm 0.09$, fails to capture the subtle transitions of composite reflectivity boundaries.

Through qualitative observations of Figure 4.12, it is evident that the base model produces output with noticeable tiling or patchiness. We suspect this artifact arises from the process of decoding each token

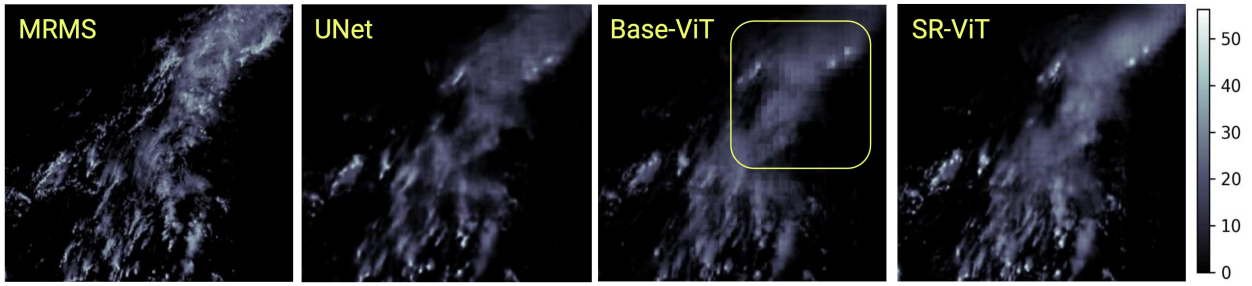



**Figure 4.12:** Observational patchiness most evident within the yellow box across model architectures.

individually, which negatively impacts the natural transition and coherence of the output. By contrast, incorporating additional convolutional layers (as done in SRViT) proves to be more effective in capturing intricate details and avoiding the tiling issues, making it a better choice for generating synthetic radar.

## 4.6 Discussion

In this chapter, we propose SRViT, a transformer-based network for generating synthetic radar from satellite observations at scale. The network functions over the entire input domain, and through multiple attention layers and heads, reconstructs the output on a per-token basis. Our analysis combines several techniques for assessing situational performance and output quality. Results show that our method improves low- and mid-value estimates of composite reflectivity while producing overall sharper predictions, outperforming prior convolutional approaches. Our case studies further validate the model's ability to accurately capture complex atmospheric phenomena, locally and across the full extent.

The primary impact of this chapter are twofold. Foremost, we provide radar reflectivity fields, efficiently and accurately, across the entire United States from satellite observations. This is a step forward in purely observational- and satellite-based machine learning products at scale. Secondly, we introduced a novel gradient-based attribution method, applicable to a breadth of transformer models for interpreting the contribution of individual tokens. This method offers a more precise understanding of token interactions and redistribution, especially within intermediate layers where traditional attention maps can become difficult to interpret. We explored this method on sample case studies and showed that it can offer additional context with valuable meteorological insights for domain experts. It is important to note that collaborations with these experts are continually needed for the greatest impact.

Despite these advancements, several challenges and limitations remain. Foremost, this method only generates 2D outputs for the use of data assimilation. However, the greatest improvements to inform NWP

includes information of the vertical profile, thus 3D fields of radar reflectivity. This would inherently change the architectural design of SRViT, and investigating computationally efficient transformer models would be favorable. Moreover, our approach is static in time, limiting the generation of future forecasts. In Chapter 5, we will move toward generative, conditional models that will enable approaches where this is desirable, albeit on a different task.

Additionally, the motivating question of whether a transformer-based network would outperform a convolutional network for image translation tasks is not fully answered. While SRViT shows improvements to low- and mid-value estimates, the high-value reflectivity was most accurately captured best by the baseline convolutional network. Furthermore, the output of SRViT, while sharper than the baseline, does not accurately represent the sharpness of ground truth radar fields. This limitation is not solely due to the architecture. Rather, we are optimizing for a weighted mean-squared error, which effectively collapses on the ensemble mean. Although the pixel-wise loss is deterministically effective, it tends to produce an average of the samples that are to be reconstructed. In Chapter 5, we explore diffusion-based methods, which build upon transformers and convolutional networks as their foundation and offer a natural improvement to these limitations. Lastly, while Token (Re)Distribution was evaluated on multiple networks and datasets, a more thorough comparison to other gradient-based measures could strengthen its utility.

# Chapter 5

# On the Dynamics of Autoregressive Generative Diffusion Models

Previous chapters have focused on discriminative models for regression, classification, and image-to-image translation tasks. The goal of these tasks were to predict or transform data based on existing observations. However, in many real-world applications, particularly in weather forecasting or climate modeling, the challenge lies in generating a sequence of outputs that evolve over time from an initial condition. This requires models capable of capturing not only the immediate relationships between data points, but also the underlying dynamics that govern long-term behavior. While transformers (Chapter 4) can be used autoregressively, the effect of doing so by themselves for image translation has output that is not very sharp or as accurate due to its training and optimization methods.

In this chapter, we shift our focus to autoregressive generative diffusion models, which are well-suited for these types of tasks. However, the dynamics of long-range timescales, particularly in the context of climate and intraseasonal predictability, have yet to be studied in depth. Understanding these dynamics not only advance our scientific understandings of both diffusion models and climate modeling, but also presents a path toward overcoming computational challenges in modeling complex temporal dependencies effectively. In particular, we study the global evolution of daily precipitation with a diffusion model (DiffObs) that we assess with domain-specific diagnostics. The model is trained to probabilistically forecast day-ahead precipitation. Nonetheless, it is stable for multi-month rollouts, which reveal a qualitatively realistic superposition of convectively coupled wave modes in the tropics. Despite secondary issues and biases, the results affirm the potential for a next generation of global diffusion models trained on increasingly sparse, and increasingly direct and differentiated observations of the world, for practical applications in subseasonal and climate prediction.

We begin with outlining the motivation for using observational data and why diffusion is well-suited to our task in Section 5.1. In Section 5.2 we detail our methodology, including a background on diffusion and our architecture with training specifics. Thereafter, we introduce the dataset for our study and experimental results in Section 5.3. Lastly, we discuss the impact and summarize our findings in Section 5.4.

Additional reading as it relates to this chapter can be found in the corresponding publication:

Stock, J., Pathak, J., Cohen, Y., Pritchard, M., Garg, P., Durran, D., Mardani, M., & Brenowitz, N. (2024). *DiffObs: Generative Diffusion for Global Forecasting of Satellite Observations*. In ICLR 2024 Workshop on Tackling Climate Change with Machine Learning, May, 2024.

## 5.1 Background and Motivation

As machine learning-driven global forecasting systems exit their infancy and move beyond weather [21–24] toward climate [29, 30] timescales, whether they can be made to generate realistic convectively coupled tropical disturbances across daily to multi-week simulations becomes an important question. Such atmospheric variability has been a longstanding challenge to capture realistically in physics-based models [13] and is still incompletely understood [14], yet regulates the subseasonal predictability of the Earth System, including important tropical to extratropical teleconnections [133].

These dynamics become especially interesting to examine in emerging autoregressive diffusion models [19, 134–136] that learn conditional probabilities and are thus well suited to the stochastic character of tropical convective dynamics, and potentially other Earth system applications. Moreover, such methods suggest that diffusion models [6, 7] do not require complete information about the atmospheric state, and thus may have the capacity to produce realistic variability even when trained on limited, direct (e.g., univariate) observations. Some recent work [137, 138] has explored diffusion models with univariate weather data, but only on short-term scales and in small spatial domains; computational advances in GPU computing allow more ambition today.

To address these challenges and explore the potential of diffusion models for long-range forecasting, we introduce a computationally ambitious, high-resolution ($0.4^\circ$) global autoregressive generative diffusion model (DiffObs) to predict the global evolution of a satellite derived precipitation product. Precipitation is observed globally via microwave sensing satellites (e.g., TRMM, [139] and GPM, [140]), and prior low-order models have proven skillful at long-range forecasts [141]. The coupling of both data and our modeling approach is pivotal to the feasibility of our current work. Using our trained model, we perform an in depth analysis of generated tropical variability on 1- to 60-day timescale using domain-informed diagnostics, discovering long-term stability with realistic variability of multiple wave structures.

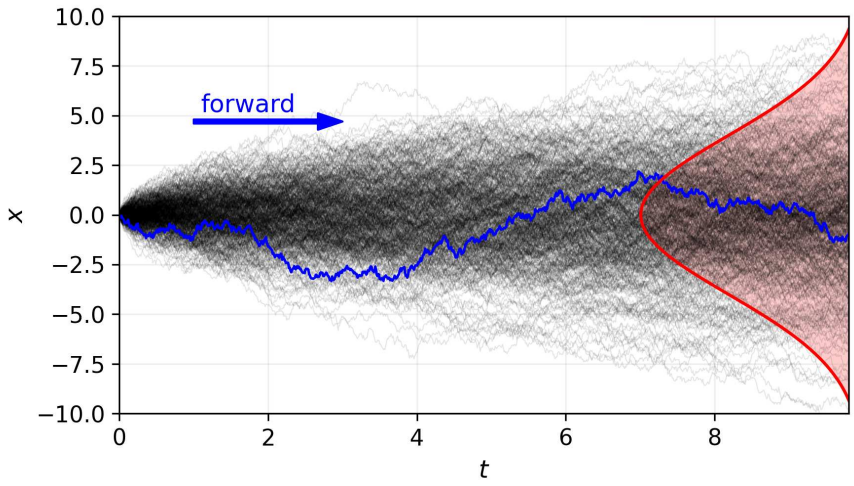



**Figure 5.1:** Evolution of a sample $x_{0\to T}$ through time following Brownian motion ($T = 1000, \mathrm{d}t = 0.01$). An example forward diffusion process is shown among 500 trials that approach a Gaussian distribution.

## 5.2 Methodology

We introduce an autoregressive diffusion model, extending the EDM architecture [6] with the objective to estimate $p\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right)$ without incorporating additional priors. Achieving this goal assumes a paired spatiotemporal relationship within the underlying distribution to effectively capture the dynamics of the system based solely on the immediate past state. In doing so, our model can rollout predictions by utilizing the estimated next step as the subsequent initial condition.

The design specifics of our model are inspired by the work of prior global diffusion-based weather forecasting models. Specifically, we build upon the work of [134], which employs a similar architecture for $\mathrm{km}$-scale downscaling. However, we avoid the use of an intermediate regression model and do not scale down the conditional inputs. Similarly, [19] present a purely autoregressive diffusion model, but train on a comprehensive state vector given from reanalysis data, where instead we directly estimate a single observational state.

### 5.2.1 Diffusion Details

Diffusion methods are defined by separate forward and backward processes as represented by stochastic differential equations (SDEs). At a high level, these processes continuously increase or decrease the noise level of an input when moving forward or backward in time, respectively. Concretely, these SDEs evolve a sample, $\mathbf{x}$, to align with some data distribution, $p$, as it propagates through time [6, 7].

As a base formulation, the forward SDE is given by

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\omega_t, \tag{5.1}$$

where $\mathbf{f}(\cdot, t) : \mathbb{R}^d \to \mathbb{R}^d$ is a vector-valued function defining the drift coefficient, $g(t) \in \mathbb{R}$ is the diffusion coefficient, and $\omega_t$ is the standard Wiener process (or Brownian motion). The solution to this SDE is the continuous accumulation of random variables $\{\mathbf{x}(t)\}_{t\in[0,T]}$.

We illustrate the forward process with Brownian motion, which has the following properties: starts with $x = 0, t = 0$, has a displacement interval $(t_0, t_1)$ that is normally distributed, and has independent intervals over non-overlapping increments. In Figure 5.1, we can see the evolution of $x_{0\to T}$ is normally distributed over multiple trials. More challenging, yet importantly, we would like to reverse this process to generate new samples initially sampled from this normal distribution of noise.

For any SDE of the form given by Equation (5.1), the closed form reverse-time SDE [142] is given by

$$\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g(t)^2\nabla_{\mathbf{x}} \log p(\mathbf{x})\right]\,\mathrm{d}t + g(t)\,\mathrm{d}\bar{\omega}_t, \tag{5.2}$$

where $\mathrm{d}t$ is a negative infinitesimal time step and $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ is the score function [143].

In practice, and in this work, we simplify Equations (5.1) and (5.2) by removing the drift coefficient, $\mathbf{f}(\cdot, t)$. The diffusion coefficient is also explicitly defined by a function of a noise scheduler, $\sigma(t)$, to prescribe a given noise level at time $t$, typically as $\sigma(t) \propto \sqrt{t}$.

This results in the *forward* (drift-removed) SDE, simplified as

$$\mathrm{d}\mathbf{x} = \sqrt{2\dot{\sigma}(t)\sigma(t)}\,\mathrm{d}\omega_t, \tag{5.3}$$

while the *reverse*-time SDE, sampled iteratively starting from $\mathbf{x}(T) \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ for a large $T \ldots 0$ (illustrated in Figure 5.2), is defined as

$$\mathrm{d}\mathbf{x} = -2\dot{\sigma}(t)\sigma(t)\nabla_{\mathbf{x}} \log p\left(\mathbf{x}; \sigma(t)\right)\,\mathrm{d}t + \sqrt{2\dot{\sigma}(t)\sigma(t)}\,\mathrm{d}\bar{\omega}_t, \tag{5.4}$$

where $\dot{\sigma}(t)$ is the time derivative of $\sigma(t)$. This is just the deterministic component representing the probability flow ordinary differential equation (ODE) with noise degradation and the noise injection component added on.

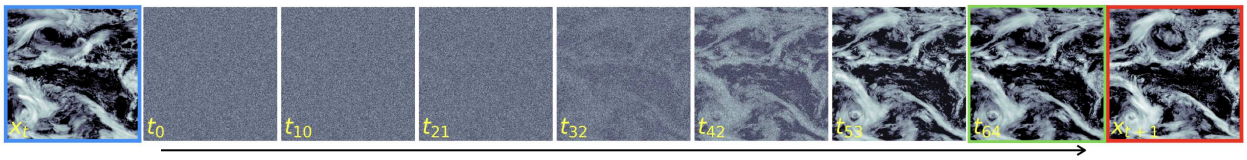



**Figure 5.2:** Reverse diffusion of a cropped sample with the input condition, individual sampling steps ($t_0 \rightarrow t_{64}$, inversely labeled), and the next time step estimate and target output.

**> vignette (3):** ***intuition of the score function***

Much of the backward process is governed by the score function, $\nabla_{\mathbf{x}} \log p(\mathbf{x})$. Intuitively, this is a gradient field to direct samples toward the underlying data distribution. In many high-dimensional machine learning problems, this function is unrealized and needs to be approximated (we will show precisely how in the coming paragraphs). However, we first derive an example using a known distribution for clarity.

Consider the multivariate Gaussian density function

$$p(\mathbf{x} \mid \mu, \mathbf{\Sigma}) = \mathcal{N}(\mu, \mathbf{\Sigma}) = (2\pi)^{-\frac{d}{2}} |\mathbf{\Sigma}|^{-\frac{1}{2}} \exp\left(-0.5(\mathbf{x}-\mu)^{\top} \mathbf{\Sigma}^{-1} (\mathbf{x}-\mu)\right), \tag{5.5}$$

where $\mathbf{x} \in \mathbb{R}^d$, $\mu$ is the mean vector, and $|\mathbf{\Sigma}|$ is the determinant of the covariance (identity) matrix. The log-density of a *mixture of two Gaussians* with equal mixing coefficients $w_1 = w_2 = 0.5$ is given by

$$\log p(\mathbf{x}) = \log\left(w_1 p(\mathbf{x} \mid \mu_1, \mathbf{\Sigma}) + w_2 p(\mathbf{x} \mid \mu_2, \mathbf{\Sigma})\right). \tag{5.6}$$

To derive the gradient of the log-density function for the mixture, we first note the gradient of the log-density for a single Gaussian with

$$\nabla_{\mathbf{x}} \log p(\mathbf{x} \mid \mu, \mathbf{\Sigma}) = -\mathbf{\Sigma}^{-1}(\mathbf{x} - \mu). \tag{5.7}$$

By exponentiation and the chain rule, we know that for any function $f(x)$, $\frac{\mathrm{d}}{\mathrm{d}x}e^{f(x)} = e^{f(x)} \cdot \frac{\mathrm{d}}{\mathrm{d}x}f(x)$. Applying this to our case gives us

$$\nabla_{\mathbf{x}} p(\mathbf{x} \mid \mu, \mathbf{\Sigma}) = p(\mathbf{x} \mid \mu, \mathbf{\Sigma}) \nabla_{\mathbf{x}} \log p(\mathbf{x} \mid \mu, \mathbf{\Sigma}) = p(\mathbf{x} \mid \mu, \mathbf{\Sigma})(-\mathbf{\Sigma}^{-1}(\mathbf{x} - \mu)). \tag{5.8}$$

Given the gradient of the log-density of the mixture is

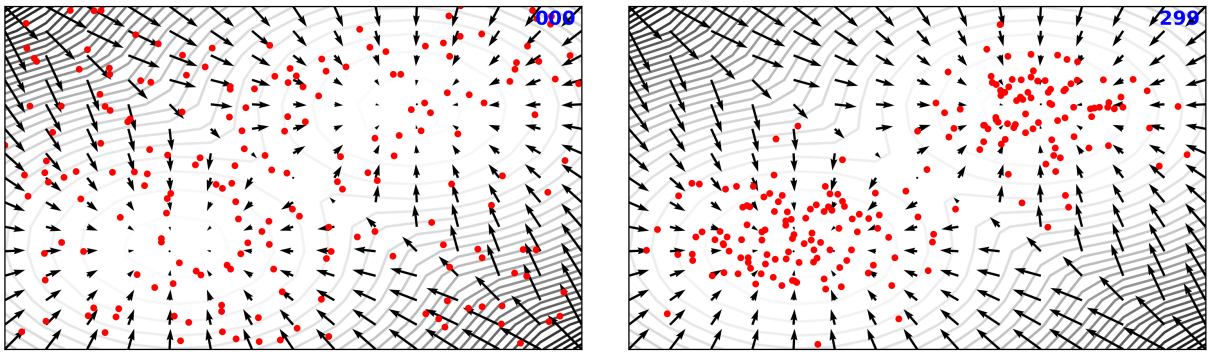
$$\nabla_{\mathbf{x}} \log p(\mathbf{x}) = \frac{\nabla_{\mathbf{x}} p(\mathbf{x})}{p(\mathbf{x})}, \tag{5.9}$$

where

$$\nabla_{\mathbf{x}} p(\mathbf{x}) = w_1 \nabla_{\mathbf{x}} p(\mathbf{x} \mid \mu_1, \mathbf{\Sigma}) + w_2 \nabla_{\mathbf{x}} p(\mathbf{x} \mid \mu_2, \mathbf{\Sigma}), \tag{5.10}$$

then by substitution, we get

$$\begin{aligned} \nabla_{\mathbf{x}} \log p(\mathbf{x}) &= \frac{w_1 p(\mathbf{x} \mid \mu_1, \mathbf{\Sigma})(-\mathbf{\Sigma}^{-1}(\mathbf{x} - \mu_1)) + w_2 p(\mathbf{x} \mid \mu_2, \mathbf{\Sigma})(-\mathbf{\Sigma}^{-1}(\mathbf{x} - \mu_2))}{w_1 p(\mathbf{x} \mid \mu_1, \mathbf{\Sigma}) + w_2 p(\mathbf{x} \mid \mu_2, \mathbf{\Sigma})} \\ &= -\mathbf{\Sigma}^{-1} \left( \frac{w_1 p(\mathbf{x} \mid \mu_1, \mathbf{\Sigma})(\mathbf{x} - \mu_1) + w_2 p(\mathbf{x} \mid \mu_2, \mathbf{\Sigma})(\mathbf{x} - \mu_2)}{w_1 p(\mathbf{x} \mid \mu_1, \mathbf{\Sigma}) + w_2 p(\mathbf{x} \mid \mu_2, \mathbf{\Sigma})} \right). \end{aligned} \tag{5.11}$$




**(a)** $\mathbf{x}_0 \sim \pi(\mathbf{x})$ **(b)** $\mathbf{x}_T \approx \mathbf{x}'$ where $\mathbf{x}' \sim p(\mathbf{x})$

**Figure 5.3:** Simulating Langevin dynamics for sampling from a mixture of two Gaussians. The red dots represent noise, **(a)** initially sampled from a normal distribution and **(b)** subsequently converging to the data distribution. The gradient field (arrows) represents the score function, while the contours are the log-density.

With this score function (Equation (5.11)), we can use Langevin dynamics [144, 145] to iteratively draw samples from $p(\mathbf{x})$. Specifically, we can use a Markov chain Monte Carlo process to obtain samples

initialized from an arbitrary prior $\mathbf{x}_0 \sim \pi(\mathbf{x})$ following

$$\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i + \epsilon \nabla_{\mathbf{x}} \log p(\mathbf{x}) + \sqrt{2\epsilon}\, \mathbf{z}_i \quad i = 0, 1, \ldots, T, \tag{5.12}$$

where $\mathbf{z}_i \sim \mathcal{N}(0, \mathbf{I})$ and $\epsilon = 0.005$. As $\epsilon \to 0$ and $T \to \infty$, the sample $\mathbf{x}_T$ is approximately equal to a typical sample drawn from the distribution $p(\mathbf{x})$. Notice the resemblance of Equation (5.12) to Equation (5.4) with the scaled score function and noise injection.

Recall our goal is to draw samples from the underlying data distribution starting from noise or a point from some arbitrary prior. Figure 5.3 illustrates this process, from the initial state to an arbitrary stopping point ($T = 300$). The contours represent the log-density (Equation (5.6)), with arrows visualizing the score function (Equation (5.11)) at different points on the grid. The initial noisy points, sampled from a normal distribution, converge to our Gaussian mixture as they follow Equation (5.12). Having a closed form solution for the score function simplifies this process, but in many scenarios, we do not know this explicitly.

For instance, consider the task of generating an image of a cat starting from a random image of a dog. If we had access to the underlying distribution of cat images, we could theoretically derive the score function and iteratively transform all dogs into cats. However, since we do not have explicit knowledge of this distribution, we instead rely on a sample population of cat images to approximate the score function. This approximation is central to the inner dynamics of diffusion models.

In diffusion models, the score function has the intriguing characteristic of not relying on the typically intractable normalization constant of the underlying base distribution $p(\mathbf{x}; \sigma)$. Exploiting this independence, we can use a denoising method defined by a neural network that minimizes the expected $L_2$ loss as a substitute of the score function, i.e., $\nabla_{\mathbf{x}} \log p(\mathbf{x}; \sigma) = \left(D_\theta\left(\mathbf{x}, \sigma\right) - \mathbf{x}\right) / \sigma^2$.

Let $D_\theta$ be a denoising model that operates on a noisy input sample given $\mathbf{x}_t \in \mathbb{R}^{c \times h \times w}$ at sample time $t$ and noise level $\sigma$, where the previous state or condition, $\mathbf{x}_{t-1}$, is concatenated channel-wise to the input. Concretely, $\mathbf{x}_t$ is the target next-day sample and $\mathbf{x}_{t-1}$ is the previous day input condition. We therefore optimize $D_\theta$ in training using

$$\min_\theta \mathbb{E}_{\mathbf{x}_{t,t-1} \sim p_{\text{data}}} \mathbb{E}_{\sigma \sim p_\sigma} \mathbb{E}_{\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})} \left[ \lambda(\sigma) \| D_\theta\left(\mathbf{x}_t + \mathbf{n}, \mathbf{x}_{t-1}; \sigma\right) - \mathbf{x}_t \|_2^2 \right], \tag{5.13}$$

where the loss weight $\lambda(\sigma) = (\sigma^2 + \sigma_{\text{data}}^2)/(\sigma \cdot \sigma_{\text{data}})^2$, the noise level $\sigma$ follows a log-normal distribution $\ln(\sigma) \sim \mathcal{N}(-1.2, 1.2^2)$, and $\sigma_{\text{data}} = 0.5$.

The denoising model, $D_\theta$, with an underlying trainable network, $F_\theta$, is preconditioned following

$$D_\theta\left(\hat{\mathbf{x}}_t, \mathbf{x}_{t-1}; \sigma\right) = c_{\text{skip}}(\sigma)\hat{\mathbf{x}}_t + c_{\text{out}}(\sigma)F_\theta\left(\left[c_{\text{in}(\sigma)}\hat{\mathbf{x}}_t, \mathbf{x}_{t-1}\right]; c_{\text{noise}}(\sigma)\right), \tag{5.14}$$

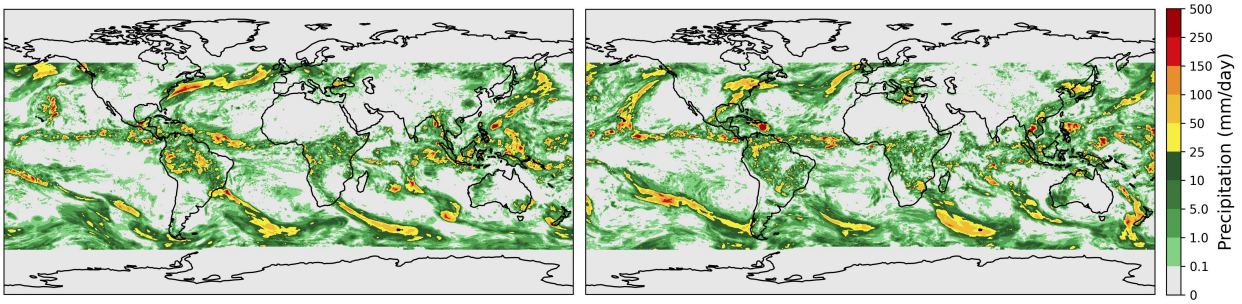
where the noisy input $\hat{\mathbf{x}}_t = \mathbf{x}_t + \mathbf{n}$, and $c_*(\sigma)$ [6, Table 1] are preconditioning variables to scale and modulate the individual components. Unlike EDM, a major change in our approach is how we concatenate the condition. Specifically, the previous timestep (condition) is concatenated channel-wise by $[\cdot]$, and $c_{\text{noise}}(\sigma)$ is an additional latent condition for $F_\theta$.

To generate samples from our model, we leverage Equation (5.4) with our trained denoising network $D_\theta$. Specifically, we iteratively solve this using the stochastic EDM sampler, which combines a second-order deterministic ODE integrator (Heun's method) with stochastic Langevin-like churn. In Figure 5.2, we illustrate this process on a cropped version of our dataset (Section 5.3.1). Much of the fine-grained detail is resolved in the final steps, while the initial steps guide the sampling direction. The final sampled output is used autoregressively as the condition for the next sample time step.

### 5.2.2 Training Details

Our experiments use the default hyperparameters outlined in [6], extending the DDPM++ UNet architecture [7], with the only deviations being the exclusion of self-attention and a reduction in model channels, specifically from $128 \rightarrow 64$. Despite the limitations imposed on the model's receptive field and its ability to capture global synoptic information without self-attention, we find the change is needed to achieve reasonable performance and training stability.

We train our 13.6M parameter model on a cluster with $256\times$ 80 GB H100 NVIDIA GPUs (32 nodes) using a global batch size of 1,024 for 12.5M total steps. End-to-end training takes 4 h wall-clock time. During generation, we sample with 64 denoising steps using the default noise levels. A single output image is fully generated in 8.5 s (unoptimized) using 4.6 GB of memory on a single GPU.




(a) Observation (b) DiffObs

**Figure 5.4:** Example 3-day rollout from DiffObs using Oct 27, 2020 as the initial condition.

## 5.3 Experiments

We begin by introducing the dataset for this study in Section 5.3.1, detailing the preprocessing and partitioning specifics, and in Section 5.3.2 we showcase and discuss our primary results.

### 5.3.1 Dataset Details

This study uses the final precipitation, half hourly Integrated Multi-satellitE Retrievals for Global Perception Measurements (IMERG) L3 Version 06B data [146, 147]. Global estimates are derived through intercalibration, morphing, and interpolating various satellite microwave precipitation and infrared retrievals, precipitation gauge analyses, and surface observations (e.g., temperature, pressure, and humidity).

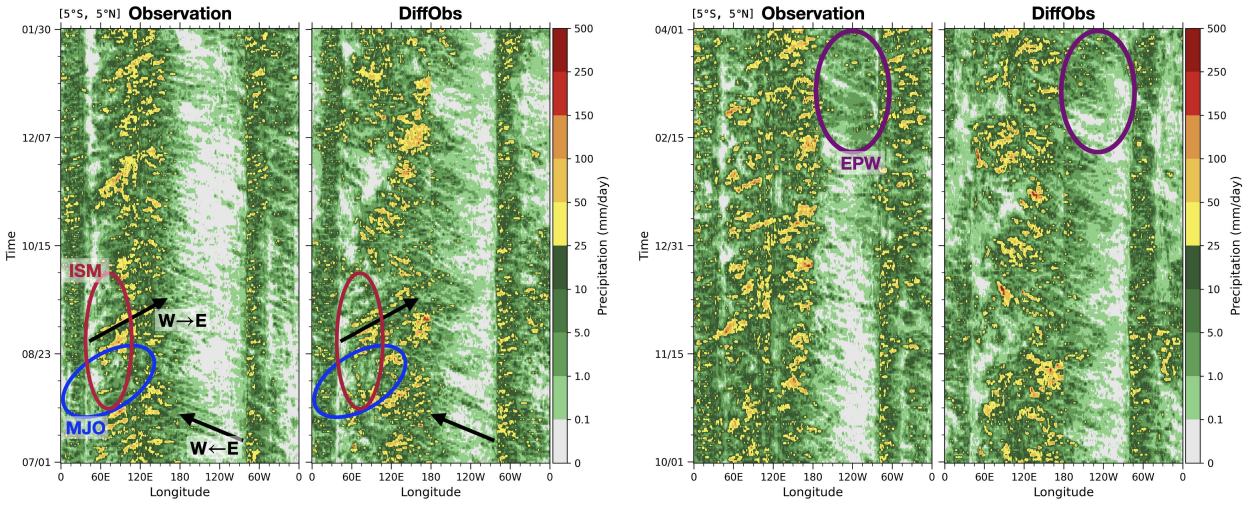
We collect data from June 1, 2000 to Sept 30, 2021 and aggregate all half hour samples for each day into an estimate of total daily precipitation (in $\mathrm{mm/d}$). Thereafter, we spatially coarsen the grid from $0.1^\circ \rightarrow 0.4^\circ$ with cropping in the meridional direction between $56.2^\circ$N and $61.8^\circ$S (296 latitudes and 900 longitudes) to avoid masking missing values at the poles. Data are partitioned to the years of 2000–2016 ($6{,}041$) for training and 2017–2022 ($1{,}729$) for testing, with the total samples in parentheses. Individual sample pairs, $(\mathbf{x}_t, \mathbf{x}_{t-1})$, are on a one-day interval with $\mathbf{x}_{t-1}$ being the condition, integrated via Equation (5.14).

Even with daily-accumulated estimates, the distribution of data is heavily right-skewed and primarily comprised of zero-valued cells with few high, yet critical precipitation values (e.g., in locations with severe weather). We therefore transform the data to be relatively Gaussian, and using statistics from the training data, normalize it between $[-1, 1]$ to align with the assumptions of diffusion models. This is done as $g(x) = 2 \cdot \ln\left(1 + x/\epsilon\right)/x_{max} - 1$, where $\epsilon = 10^{-4}$ and $x_{max} = 17.35$ as found in the transformed data. Computing $g^{-1}(x)$ on model output returns the data to its original units.

### 5.3.2 Main Findings

Using our trained model, we can generate forecasts for arbitrary lead times and leverage its inherent probabilistic nature to create an ensemble of forecasts from any initial condition. A representative member of an ensemble is shown in Figure 5.4, featuring a 3-day rollout initialized with a sample from Oct 27, 2020, to illustrate example output. While exact features should not be expected to match perfectly due to atmospheric chaos, it is notable that the forecast maintains qualitative sharpness, addressing concerns observed in deterministic convolutional [5, 148] and transformer models (Chapter 4) with pixel-wise loss functions, and accurately captures the structure of atmospheric conditions, including many high-valued precipitation events.

Next, we shift our focus to evaluate long, multi-month rollouts to study multi-scale generated atmospheric variability near the equator. Figures 5.5 and 5.6 illustrate our key findings and capabilities.




**(a)** Summer to Winter

**(b)** Boreal Winter

**Figure 5.5:** Hovmöller diagrams of observations (left) and DiffObs output (right, generated autoregressively) between $5°$N and $5°$S for case studies initially conditioned on **(a)** July 1, 2020 and **(b)** Oct 1, 2019. Individual colors correspond to the wave propagation directions (W↔E), Indian Summer Monsoon (ISM), Madden–Julian oscillation (MJO), and East Pacific Wavetrain (EPW).

#### Hovmöller Diagrams

We first compare how convectively coupled equatorial waves (averaged between $5°$S and $5°$N) propagate through longitude and time relative to observations with a Hovmöller Diagram [149]—a preferred domain diagnostic—in Figure 5.5. This diagram is found by concatenating the vector-valued latitude average (between

$5°$N and $5°$S) of samples temporally over a long enough period. Qualitative comparisons are made between the observational test data and DiffObs generated rollouts, initially conditioned on the single test observation. In Figure 5.5, this is the sample corresponding to the lower left timestamp.

A reassuring superposition of eastward- and westward-propagating tropical disturbances are generated at appropriate longitudes, modulated by a large-scale envelope of slow, eastward moving variability characteristic of the Madden–Julian oscillation [MJO, 150], at its expected location spanning the Indian Ocean ($50$–$60°$E) to West-Central Pacific ($180°$E). Encouraging East Pacific Wavetrain [EPW, 151] variability is also found, alongside some artifacts, such as a dateline discontinuity at 180°E/W and a bias towards too much time-mean precipitation generated between (additional comments in Section 5.3.4).

A single rollout is illustrated within one diagram (Figures 5.5a and 5.5b), and does not consider the probabilistic nature of multiple runs. That is, we only assess a single ensemble member. This makes it difficult to conclusively say if our observational results are merely circumstantial or not. However, the results of these diagrams alone are very promising and warrant additional, more comprehensive analyses over larger quantities of generated output.

**Wheeler–Kiladis Space-Time Spectra**

Our second analysis (Figure 5.6) provides more extensive evaluations by examining the dispersion relationships revealed in the wavenumber–frequency domain, following the methods in [152, 153]. In their seminal work, [152] showed how fast and slowly oscillating atmospheric waves, some of which arise in idealized theories [154], can be observed through spectral analysis of satellite-observed outgoing longwave radiation, including the Madden–Julian oscillation. This form of two dimensional spectral analysis has became one of the fundamental techniques for evaluating numerical models' representation of tropical waves that regulate predictability on synoptic to subseasonal timescales.

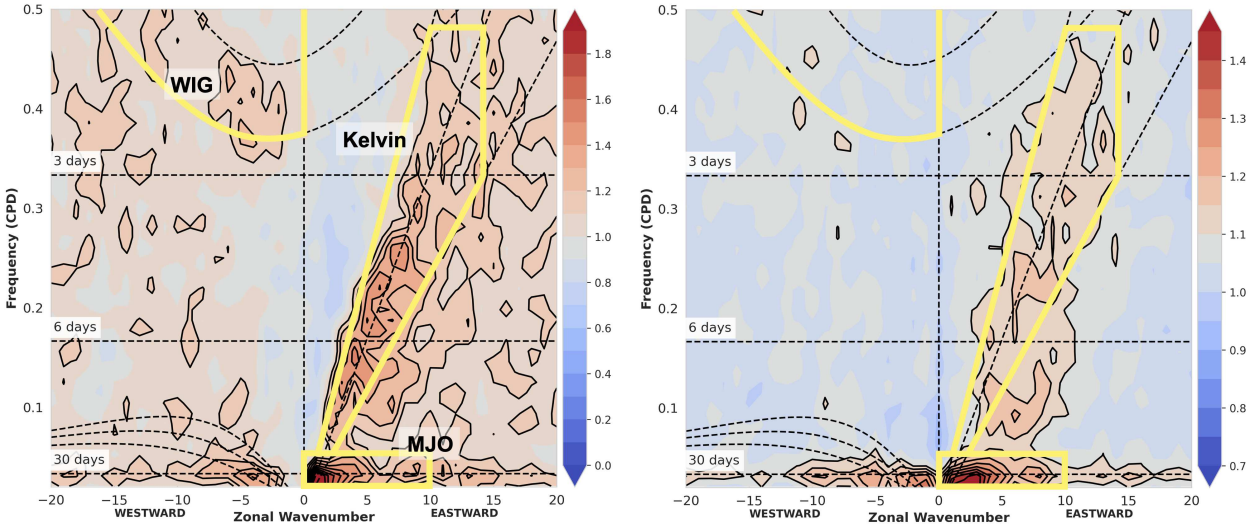
Experimentally, we generate $80\,\mathrm{yrs}$ of data on one-day intervals, initially conditioning $1\,\mathrm{yr}$ rollouts on Jan 1 for years 2017–2021 and sample with perturbed noise, creating an ensemble of predictions. Temporally concatenating the results within $15°$N/S of the equator, we perform spectral analysis to construct a Wheeler–Kiladis diagram, utilizing $96\,\mathrm{d}$ windows with a $65\,\mathrm{d}$ overlap to isolate the significant spectral peaks. This is explicitly done by (**1**) computing the 2D Fourier transform in time and longitude dimensions of the Hovmöller, (**2**) decomposing the result into the symmetric component, (**3**) calculating the power spectra, and (**4**) removing background signal (red-noise) for clarity.

As a baseline, Figure 5.6a illustrates the equatorially symmetric signal-to-noise for observations from the 5 years of test data. Key features, familiar to domain scientists, include the dominant power and east-to-west asymmetry on intraseasonal timescales (periodicity longer than 30 days), notably highlighting the MJO, as well as elevated power for an eastward-propagating convectively coupled Kelvin wave [155], spanning wavenumbers 1–14 with periods ranging from 2.5 to 25 days, and exhibiting a quasi-linear dispersion relationship [156].

Encouragingly, DiffObs reproduces both of these dominant observed features (Figure 5.6b): the spectral signal of generated power also occurs on intraseasonal timescales, i.e., timescales longer than 30 day periodicity, and across a band of spatial (zonal) wavenumbers 0–9 consistent with a planetary scale, eastward moving mode of variability. Meanwhile, on shorter timescales, the model also generates a moist Kelvin wave spectral power maximum with a qualitatively correct dispersion relationship. Despite other imperfections, such as a tendency for the model to generate too much variability at all wavelengths (Figures 5.7d and 5.7f), and an under-representation of power within westward moving tropical wave classes, these are impressive preliminary results.

Computing the Wheeler–Kiladis diagram reveals that DiffObs captures many, if not all, of the predominant observed modes and tropical wave signals. This could be viewed as a surprising property of a machine learning model trained only on a single variable, given that dynamical theories of such waves encompass several atmospheric state variables operating in concert—e.g., the vorticity and divergence of the horizontal winds are required for modeling in the equatorial Rossby and Kelvin waves, respectively, while the MJO requires complex nonlinear advection and moisture effects.

Our main finding is the discovery of Kelvin wave and MJO spectral signals within the signal-to-noise ratio of the equatorially-symmetric component of the space-time spectra (see Figure 5.6b). However, this is only one view of the analysis, and we glean more details from the additional components of the analysis. Specifically, in the antisymmetric component of observations (Figure 5.7a), there is evidence of a mixed Rossby-gravity (MRG) and eastward inertio-gravity (EIG) waves that are not apparent in Figure 5.7b. In Figures 5.7d and 5.7f, we show the intermediate computations, where the background spectra of model output is more similar to the raw power spectra than that of the observations. This suggests that DiffObs has a strong background signal that is similar to red-noise and that overall DiffObs generates too much variance.




**(a)** Observation **(b)** DiffObs

**Figure 5.6:** Symmetric / Background Wheeler–Kiladis space-time spectra between 15°N and 15°S. The individually highlighted regions correspond to where the Madden–Julian oscillation (MJO), Kelvin and westward inertio-gravity (WIG) waves are expected to be found.

### 5.3.3 Short-Term Predictability

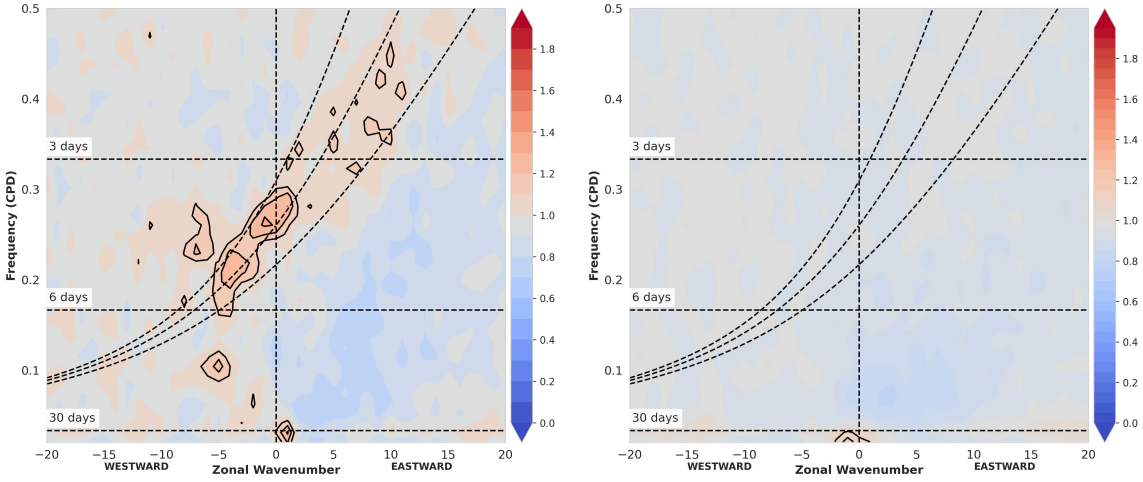
While short-term predictability is not the focus of this work, we find it is important to assess for comprehension and to be relatable to prior work. As such, we evaluate quantitative model predictability using a modified root-mean-squared error (RMSE) from [19] with assessments relative to forecasts derived by the observations, and the Fractions Skill Score (FSS) [157, 158]. While the Continuous Ranked Probability Score (CRPS) and Brier Skill Score (BSS) are common metrics to evaluate forecast performance, we currently do not have other datasets (such as the IFS or ERA5 [159]) to make for a meaningful comparison.

We define FSS for a given neighborhood size as

$$\mathrm{FSS} = 1 - \frac{\frac{1}{n}\sum_n \left(P_y - P_t\right)^2}{\frac{1}{n}\left[\sum_n P_y^2 + \sum_n P_t^2\right]}, \tag{5.15}$$

where $P_y$ is the forecast fraction and $P_t$ is the target observed fraction (exceeding a certain threshold), and $n$ is the number of spatial windows over the domain. We define the ensemble mean RMSE similar to [19] as

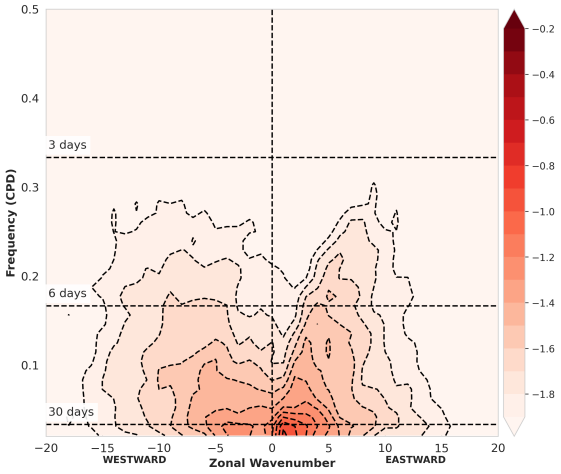
$$\mathrm{RMSE} = \sqrt{\frac{1}{M}\sum_m \frac{1}{|G|}\sum_i a_i \left(t_{i,m} - \bar{y}_{i,m}\right)^2}, \tag{5.16}$$



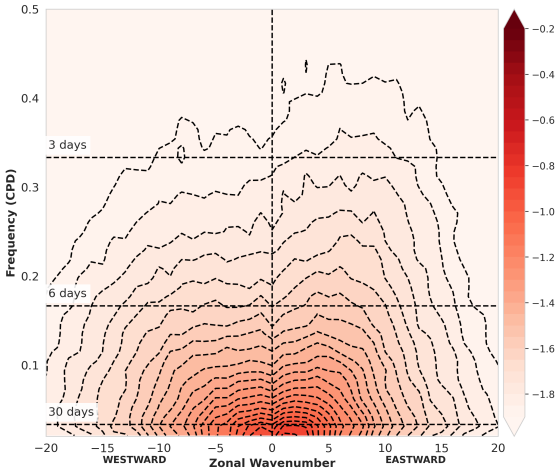

**(a)** Antisymmetric / Background (**Obs**)



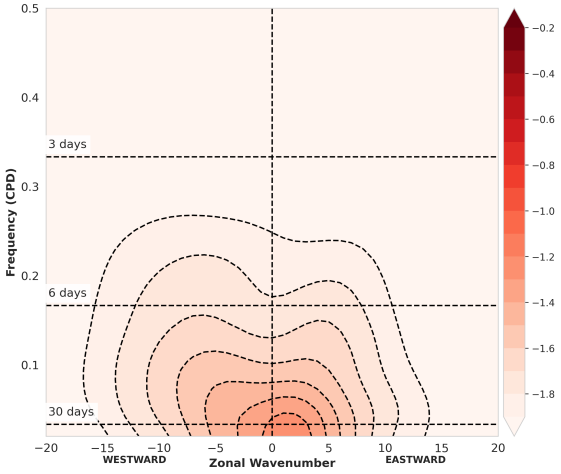

**(b)** Antisymmetric / Background (**DiffObs**)



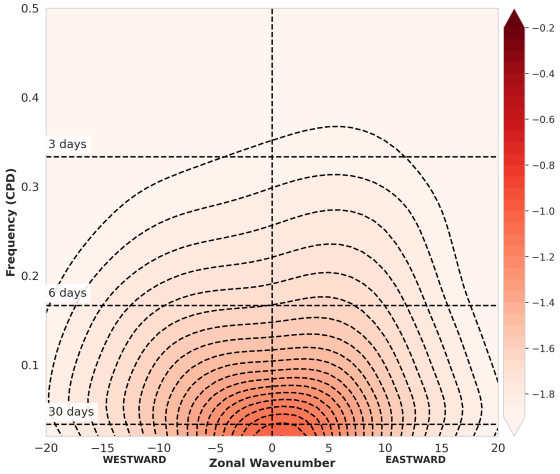

**(c)** Symmetric Raw Log Power (**Obs**)




**(d)** Symmetric Raw Log Power (**DiffObs**)

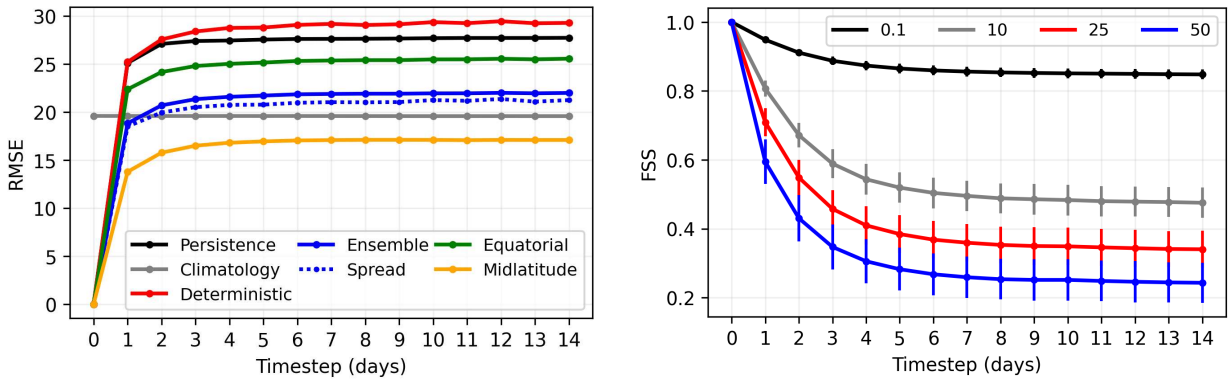



**(e)** Background Log Power (**Obs**)




**(f)** Background Log Power (**DiffObs**)

**Figure 5.7:** Additional Wheeler–Kiladis components and power spectra of observations (left column, **(a, c, e)**) and model output (right column, **(b, d, f)**) that support Figure 5.6 and Section 5.3.2.




**(a)** RMSE

**(b)** FSS

**Figure 5.8:** Short-term predictability for 14-day forecasts, initiated from each day between Jan 1, 2020 and Sept 15, 2021 with 5 ensemble members. Panel **(a)** shows the RMSE and **(b)** shows the ensemble mean FSS with different thresholds ($\mathrm{mm/d}$) at a $8.4°$ neighborhood (21 pixels).

where $y^n_{i,m}$ denotes the $n \in N$ ensemble for the $m \in M$ forecast for a lead time at a latitude and longitude indexed by $i \in G$, $t_{i,m}$ is the target observation, and $\bar{y}_{i,m} = 1/N \sum_n y^n_{i,m}$ is the ensemble mean of model predictions. We use a latitude weighting derived from the area mean on a sphere, normalized to have unit mean as defined by

$$a_i = \frac{\cos\left(\mathrm{lat}\left(i\right)\right)}{\frac{1}{N_{\mathrm{lat}}} \sum_j \cos\left(\mathrm{lat}\left(j\right)\right)}. \tag{5.17}$$

We compare individual scores to the *persistence forecast* taken as the initial condition repeated over the forecast duration (14-days) as well as *climatology* found by a two week average window of mean daily conditions for years 2000–2022. Additionally, we compute the ensemble mean errors individually at the midlatitudes (between 30°N/S and 60°N/S) and tropics (between 23.5°N and 23.5°S) without any latitude weighting. We also show the deterministic error and the ensemble spread as the square root of $(n+1)/n$ times the average over all forecasts of the ensemble variance.

Figure 5.8a shows the forecast errors plateau quickly after a 3-day lead time, yet there is consistency with the persistence and deterministic errors approaching the ratio of $\sqrt{2}$ with lead time for climatology and the ensemble, respectively. Even though the ensemble error is better than persistence, given this consistency and that the model is under-dispersive (spread $<$ error), we can deduce that the ensemble mean is relatively poor. By computing errors at varying latitudes, we see the greatest error with the estimations in the tropics, where there is also a bias toward high precipitation values.

In evaluating our model using FSS, various neighborhood sizes were considered (not shown within), revealing that the $8.4°$ neighborhood size effectively captures large-scale atmospheric structure. The outcomes

of experimenting with different thresholds are presented in Figure 5.8b. Notably, lower thresholds have higher skill, which continuously decreases with high-valued estimates, and the highest skill is at early lead times (out to 5 days), where skill plateau thereafter as seen with RMSE.

### 5.3.4 Issues of the Dateline Discontinuity

In our experiments, we observe a dateline discontinuity at $180°$E/W, evident down the center of the Hovmöller in Figure 5.5 (right). Ideally, we would like our model to be consistent around the globe and allow for periodic wave propagation. We aim to address this by modifying our network architecture and by including additional conditions, as outlined in Section 5.3.4. However, we find the changes to be suboptimal, showing in Section 5.3.4 further inconsistencies and worse performance relative to our baseline model (DiffObs).

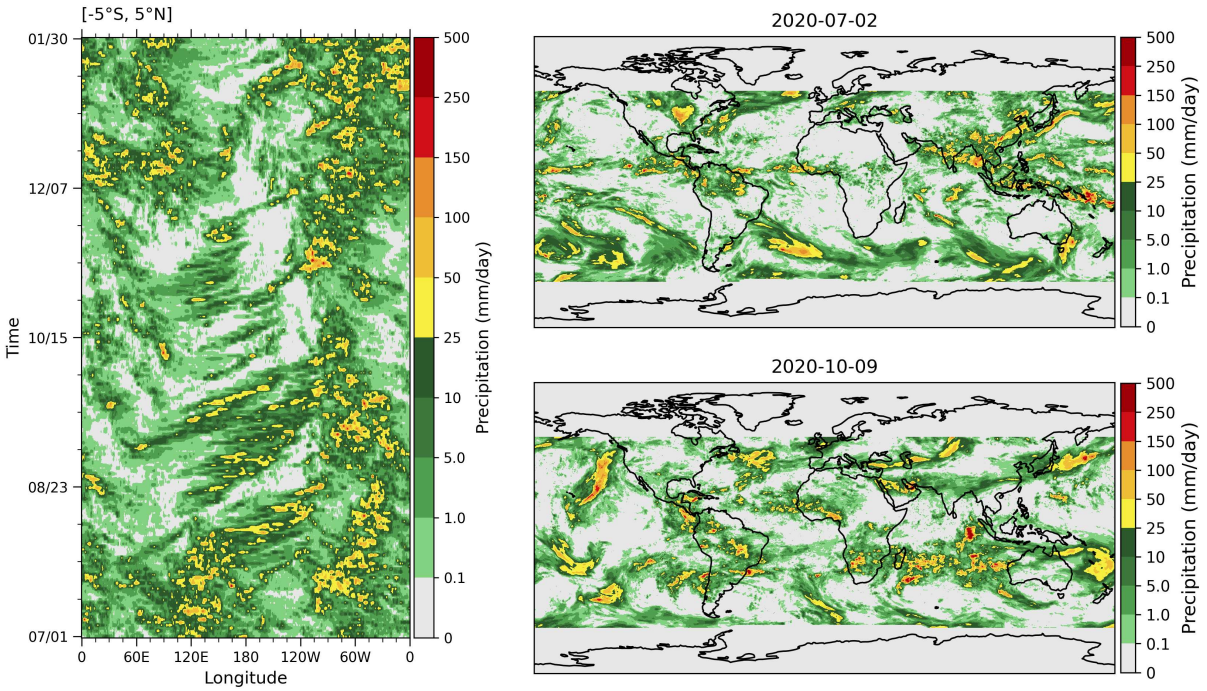
#### Training Modifications

For our network to have rotational equivariance, we modify the convolutions to use circular-padding in the zonal direction and zero-padding in meridional direction (due to the cylindrical structure of our data). This effectively removes any spatial bindings given by the dateline. As such, we include a two-channel static condition (concatenated channel-wise to the existing condition) of $\cos(\text{lon})$ and $\sin(\text{lon})$, repeated over the meridional directions. These additional conditions, spatially aligned with the input, *should* maintain spatial coherence.

In addition to the padding and coordinate conditions, we also include the zonal average of the cosine of solar zenith angle as a function of the condition date and latitudes to account for temporal variability. We compute this for each latitude, $\phi$, at time $t$ as,

$$\cos\theta_s = \sin\phi\sin\delta + \cos\phi\cos\delta\cos h, \tag{5.18}$$

where $\delta$ is the declination of the sun and $h$ is the hour angle. Given that our data is daily-accumulated, we integrate time by zonally averaging at UTC+0 and repeat the value for each latitude. The result is again concatenated channel-wise and *should* be effective to provide seasonal cycle conditioning. It is important to note that during training, we use the date associated with the condition (i.e., the previous timestep), iterating $\cos\theta_s$ in time during a rollout.




**(a)** Hovmöller

**(b)** Individual Output Samples

**Figure 5.9:** Experimental DiffObs output aimed to resolve the dateline discontinuity (Section 5.3.4), generated autoregressively when initially conditioned on July 1, 2020. Panel **(a)** is the Hovmöller between $5°$S to $5°$N and **(b)** are individual output samples with their corresponding steps shown by the date.

### Experimental Results

We train our updated model with the same hyperparameters and training specifics detailed in Section 5.2 and repeat similar evaluations from Section 5.3. While the periodicity is preserved across the dateline, we find the results to inadequately represent the atmospheric dynamics. The most salient representation of this is illustrated in Figure 5.9a when comparing to observation (Figure 5.5, left). Notably, no landmasses are identified, eastward-propagating waves traverse the dateline, and oscillating wave signals are not captured. While it is not abundantly clear as to why these modifications yield worse results, we note that there should be careful considerations when iterating on future work.

## 5.4 Discussion

We have demonstrated an autoregressive, univariate diffusion model for predicting daily-accumulated precipitation based only on the previous day's data. The model produces stable long rollouts that exhibit a realistic spectrum of tropical wave modes (confirmed with cross-spectral analysis), including the Madden–Julian oscillation, whose variance dominates on intraseasonal timescales and is notoriously difficult to

simulate realistically in physics-based models [14] as well as convectively coupled moist Kelvin waves with approximately correct dispersion relationships. Interestingly, DiffObs does not exhibit high skill in short-term predictability, but our results affirm its nontrivial ability to autoregressively model complex multi-scale patterns over extended temporal scales.

This result holds significant promise for the next generation of global diffusion models, particularly in leveraging increasingly sparse and differentiated observations of the world for subseasonal and climate prediction applications. More importantly, the discovery of stable long-range dynamics emphasizes the broader potential of generative models, extending beyond weather and climate. While single-step diffusion models may not always excel at short-term predictability, we find that they demonstrate an impressive ability to capture patterns that extend hundreds or even thousands of steps into the future. We believe that exploring approaches that explicitly model temporal dependencies, such as conditioning on multiple timesteps to estimate $p\left(\mathbf{x}_t|\mathbf{x}_{t-n}, \ldots, \mathbf{x}_{t-1}\right)$, could help capture more accurate and correlated patterns over these long periods.

Several auxiliary challenges had to be addressed to ensure the stability of our model, particularly when working with high-resolution imagery. As image size increases, pixel redundancy introduces noise, making it difficult to maintain performance. We hypothesized that larger batch sizes would help mitigate this issue, supported by empirical evidence from [6, 160] that indicate performance improves with increased batch size. However, our initial experiments were constrained by limited computational resources, leading us to explore three alternative strategies.

First, we downsampled and interpolated the samples to collapse the spatial domain. While this allowed for larger batch sizes and accurate predictions, our goal was to retain high-resolution outputs. We also experimented with wavelet transforms (connecting this work to Chapter 2), to reduce the spatial dimension. Specifically, inspired by [161, Figure 6], we applied an invertible 5/3 linear wavelet transform to the first and last layers of the denoising network. Similarly, we explored using the scattering transform, learning its inverse in the output layer. Although we found the downsampling methods to be effective for prototyping and iterating quickly, the wavelet-based approaches introduced instability and required more hyperparameter tuning. Ultimately, scaling compute to allow for a larger global batch size produced the most successful results.

Additional challenges and biases also need to be acknowledged. Cross-spectral analysis reveals that not all observed wave modes are captured by our model, particularly westward inertio-gravity and mixed

Rossby–gravity waves, which only appear in the symmetric and antisymmetric observational components, respectively. Another limitation is the dateline discontinuity in our primary model, and our attempts to resolve this further emphasize the challenge. Lastly, the IMERG data we use is a level 3 product that integrates multiple data sources, including ERA5, and moving toward lower-level satellite products would be ideal. Despite these issues, we believe our work takes an important step away from reanalysis, though integrating models like the IFS and ERA5 could further improve evaluations of observation-based modeling.

# Chapter 6

# Conclusion and Future Work

This dissertation presents a range of complementary methods tailored to various applications and challenges in computer and atmospheric science. The design of these methods built upon one another and were motivated by the inherent structure of the data and task at hand. By integrating classical vision techniques, various attention methods, and generative models we have demonstrated how machine learning can improve predictions of atmospheric phenomena, synthesize observational data, and capture long-range, dynamic atmospheric variability.

In earlier chapters, wavelet- and scattering-based networks were shown to effectively capture multi-scale patterns, leading to improvements in detecting gravity waves, estimating the intensity of tropical cyclones, and predicting the occurrence of lightning from satellite observations. These advancements were made possible through our parameterized wavelet layer and channel-separated scattering attention scheme. Subsequently, our model of memory-based sequential attention, built using a transformer encoder and inspired by cognitive psychology and neuroscience, showed how a stochastic policy can guide attention to improve both performance and the potential for model explainability. By comparing sample trajectories with traditional saliency maps, we gleaned new insights to the indicators of climate change.

Transitioning to image-to-image translation, we explored the capabilities of transformers in generating high-resolution, synthetic radar fields, showing improvements to sharpness and accuracy over a convolutional approach. Resulting from this, we introduced an explainability method to show how input tokens are combined to help guide the intuition for domain experts. Lastly, we designed an autoregressive generative diffusion model to produce long rollouts of precipitation fields, discovering, through the use of domain-specific diagnostics, stable and realistic tropical variability over intraseasonal timescales. Moreover, showcasing the capability of single-step diffusion models for capturing internal data-driven dynamics.

There is no single method that can be applied to every application; each has its own strengths and weaknesses that should be considered. In Section 6.1, we highlight and compare each method, and in Section 6.2, we discuss the future directions of this work and offer perspectives on the field at large.

## 6.1 Comparison of Methods

The WaveNet model (Section 2.2) is especially effective when we have some prior knowledge of underlying periodicity. Its interpretability stems from learning dominant frequencies directly from the data with parameterized wavelets that can be evaluated post hoc to derive insights from. Furthermore, this approach significantly reduces the total number of parameters in comparison to more complex architectures. Building on this, the scattering network (Section 2.3) learns to attend to the coefficients from predefined wavelets on a per channel basis. Its performance is robust when training data is scarce and can provide expressive explanations for each of these channels. By contrast, the transformer and diffusion models we introduce in Chapters 4 and 5, albeit for different tasks, require a significant amount of data to perform well.

Under similar regression and classification tasks, our Memory-Based Sequential Attention model (Chapter 3) differs from those above, specifically in how the input space is restricted to a sequence of glimpses. This trajectory is found through a control strategy learned with reinforcement learning that jointly optimizes for data likelihood. As such, the trajectories are stochastically sampled and can vary slightly between runs. This probabilistic variability is also seen with our diffusion model in Chapter 5. Sequential attention is well-suited for tasks where there are clear, informative features or for discovering new features from the data. Moreover, by reducing the input to a smaller set of features, we can use the internal attention weights to glean insights to how these features are used to form a prediction, similar to the scattering network in Section 2.3.

Both sequential attention and our vision transformer for image-to-image translation (Chapter 4) leverage self-attention. In the former, we use a transformer encoder in the memory module to process candidate image patches. However, our vision transformer applies attention to every patch in an image, akin to traditional vision transformers but adapted for image translation. This approach is particularly useful when dealing with high-resolution imagery, where sparse, large-scale spatial dependencies often arise. These dependencies cannot be fully captured within the receptive fields of most convolutional networks. In such cases, our model provides an effective solution. Additionally, we derive an attribution method termed Token (Re)Distribution, which can be applied to other transformer-based models to interpret how information flows into specific tokens; it is not specific to image-to-image translation. However, the training objectives of deterministic image translation, such as SRViT, are limited in their ability to capture the inherent variability and uncertainty of real-world data.

By adopting a diffusion-based approach that learns the underlying data distribution, we can probabilistically generate more diverse, and fixable outputs—although not entirely unbiased. In Chapter 5, we introduce

DiffObs, which successfully captures realistic atmospheric variability on longer timescales in an autoregressive setting, but still struggles with short- to medium-range precipitation forecasts. Diffusion models are well-suited for generating an ensemble of predictions, often outside the mean of the data, with more realistic and sharper features than deterministic approaches. This may be more favorable for generating synthetic imagery, as opposed to the kind of deterministic approaches introduced in Chapter 4.

Each of our proposed methods offer distinct advantages, with some better suited for small, structured datasets and others capable of handling large, unstructured data. The trade-offs between data characteristics, learning objectives, and architectural complexity must be carefully considered based on the specific application. Selecting the right method depends on the balance between simplicity, interpretability, and the ability to model complex patterns, whether dealing with sparse observations or generating high-resolution outputs.

## 6.2 Future Work

This dissertation has explored a range of machine learning methods for atmospheric science, spanning both predictive tasks and generative modeling. However, several exciting directions remain to further improve these approaches, particularly around model explainability, observation-based modeling, and hybrid approaches.

In Chapter 2, we embed data-specific characteristics directly into the architecture and training objectives, where we saw an improvement to performance and the potential for explainability (requiring further human intervention and interpretation). An extension of this leans into hybrid modeling, where we could incorporate parameterized physical transformations, much like the wavelet transform. This could improve the interpretability of complex models and reduce their reliance on large datasets by including additional domain knowledge into the learning algorithm. Additionally, we could adapt our model of sequential attention in Chapter 3 by incorporating spatial attention from Chapter 2 to inform region selection based on saliency. Moreover, while this model is currently designed in the spatial domain, extending it to higher dimensions could allow for adaptions to vertical or temporal domains.

As it relates to Chapters 2 and 3, the domain-specific tasks are mostly demonstrations and future work should include additional collaborations with domain experts to strengthen the work. Iterating with experts would allow for a more thorough evaluation of both results and model correctness. Furthermore, the algorithms themselves have the potential to be refined to better incorporate domain knowledge. As a result, we could not only yield new scientific insights but also improve the potential for more trustworthy use.

Much of this dissertation has relied on observations and observational-derived products. Moving beyond reanalysis and numerical weather prediction has great potential, particularly in reducing reliance on organizational data pipelines. In Chapter 5, we showed the efficacy of observational data for subseasonal predictability, but future work should explore the potential of even lower-level satellite products. A natural extension would be to adopt a diffusion model, such as that in Chapter 5, for the application in Chapter 4: directly modeling radar from satellite observations. The overall contributions we have made in observation-based modeling hold promise but they remain largely underexplored, especially for global forecasting.

Our results also suggest that generative diffusion models can serve as a foundation for learning dynamical priors, allowing the models to capture more realistic long-term behaviors. To further improve modeling efforts, we see potential in incorporating self-supervised learning at test time to allow for continuous adaptations to new, unseen environments. By updating the model in an online setting using the underlying dynamics from unseen data, we could potentially maintain high performance under distribution shifts. This could significantly improve weather forecasts and offer more flexible and accurate predictions in scenarios where current models struggle, such as abrupt changes or rare meteorological events.

## 6.3 Applications

Altogether, the machine learning methods introduced in this dissertation, though predominantly applied to atmospheric science, have broader potential in various domains. The adaptive wavelet network (Chapter 2) could be extended to improve vertical profiles of temperature and moisture from numerical weather prediction models [162, 32] or with brain-computer interface tasks, such as detecting mental activity or event-related potentials from electroencephalograms [163]. In both cases, the inherent one-dimensional structure of the data make these applications particularly well-suited.

Our model of sequential attention (Chapter 3) could also be used to assess and discover cloud patterns associated with tropical cyclones, offering insights and guidance into intensification patterns, similar to the Dvorak technique [164]. This is especially relevant given the increased risk of tropical cyclone intensification [165, 166]. Beyond meteorology, our model could be adapted for medical diagnoses and support radiographers in X-ray interpretations [167, 168].

Additionally, our token attribution method (Chapter 4) could complement other, emerging explainability methods [169, 129]. This need not be constrained to the vision domain, but could be extended for use in explaining the flow of information in large language models. Finally, our diffusion model (Chapter 5) could

be adapted to integrate multivariate data or even push toward more sparse observations. In oceanography, for example, a global ocean model trained on Argo data [170] could extend predictions to cover vertical ocean dynamics.

For each of the example applications discussed above, it is critically important to continue to collaborate with domain experts. Interdisciplinary research fosters not only novel discoveries but also ensures that algorithmic approaches remain well aligned with the practical challenges of the domain.

# Bibliography


[1] Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences*, 121(27):e2311878121, 2024.

[2] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12104–12113, 2022.

[3] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2020.

[4] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6228–6237, 2018.

[5] Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, 2021.

[6] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

[7] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[8] Amy McGovern, Imme Ebert-Uphoff, David John Gagne, and Ann Bostrom. Why we need to focus on developing ethical, responsible, and trustworthy artificial intelligence approaches for environmental science. *Environmental Data Science*, 1:e6, 2022.

[9] Marcin Andrychowicz, Lasse Espeholt, Di Li, Samier Merchant, Alexander Merose, Fred Zyda, Shreya Agrawal, and Nal Kalchbrenner. Deep learning for day forecasts from sparse observations. *arXiv preprint arXiv:2306.06079*, 2023.

[10] Peter Manshausen, Yair Cohen, Jaideep Pathak, Mike Pritchard, Piyush Garg, Morteza Mardani, Karthik Kashinath, Simon Byrne, and Noah Brenowitz. Generative data assimilation of sparse weather station observations at kilometer scales. *arXiv preprint arXiv:2406.16947*, 2024.

[11] Julia Slingo and Tim Palmer. Uncertainty in weather and climate prediction. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 369(1956):4751–4767, 2011.

[12] Flavio Lehner, Clara Deser, Nicola Maher, Jochem Marotzke, Erich M Fischer, Lukas Brunner, Reto Knutti, and Ed Hawkins. Partitioning climate projection uncertainty with multiple large ensembles and cmip5/6. *Earth System Dynamics*, 11(2):491–508, 2020.

[13] David A Randall. Beyond deadlock. *Geophysical Research Letters*, 40(22):5970–5976, 2013.

[14] C. Zhang, Á. F. Adames, B. Khouider, B. Wang, and D. Yang. Four theories of the madden-julian oscillation. *Reviews of Geophysics*, 58(3):e2019RG000685, 2020. doi: https://doi.org/10.1029/2019RG000685. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019RG000685. e2019RG000685 2019RG000685.

[15] Amy McGovern, Julie Demuth, Ann Bostrom, Christopher D Wirz, Philippe E Tissot, Mariana G Cains, and Kate D Musgrave. The value of convergence research for developing trustworthy ai for weather, climate, and ocean hazards. *npj Natural Hazards*, 1(1):13, 2024.

[16] Rebecca E Morss, Heather Lazrus, and Julie L Demuth. The "inter"' within interdisciplinary research: Strategies for building integration across fields. *Risk Analysis*, 41(7):1152–1161, 2021.

[17] Lander Ver Hoef, Henry Adams, Emily J King, and Imme Ebert-Uphoff. A primer on topological data analysis to support image analysis tasks in environmental science. *Artificial Intelligence for the Earth Systems*, 2(1), 2023.

[18] Longcai Zhao, Qiangzi Li, Yuan Zhang, Hongyan Wang, and Xin Du. Integrating the continuous wavelet transform and a convolutional neural network to identify vineyard using time series satellite images. *Remote Sensing*, 11(22):2641, November 2019.

[19] Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Timo Ewalds, Andrew El-Kadi, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, and Matthew Willson. Gencast: Diffusion-based ensemble forecasting for medium-range weather. *arXiv preprint arXiv:2312.15796*, 2023.

[20] Matthias Karlbauer, Nathaniel Cresswell-Clay, Dale R Durran, Raul A Moreno, Thorsten Kurth, Boris Bonev, Noah Brenowitz, and Martin V Butz. Advancing parsimonious deep learning weather prediction using the healpix mesh. *Journal of Advances in Modeling Earth Systems*, 16(8):e2023MS004021, 2024.

[21] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

[22] Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Pangu-weather: A 3d high-resolution model for fast and accurate global weather forecast. *arXiv preprint arXiv:2211.02556*, 2022.

[23] Kang Chen, Tao Han, Junchao Gong, Lei Bai, Fenghua Ling, Jing-Jia Luo, Xi Chen, Leiming Ma, Tianning Zhang, Rui Su, et al. Fengwu: Pushing the skillful global medium-range weather forecast beyond 10 days lead. *arXiv preprint arXiv:2304.02948*, 2023.

[24] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Alexander Pritzel, Suman Ravuri, Timo Ewalds, Ferran Alet, Zach Eaton-Rosen, et al. Graphcast: Learning skillful medium-range global weather forecasting. *arXiv preprint arXiv:2212.12794*, 2022.

[25] Jamin K Rader, Elizabeth A Barnes, Imme Ebert-Uphoff, and Chuck Anderson. Detection of forced change within combined climate fields using explainable neural networks. *Journal of Advances in Modeling Earth Systems*, 14(7):e2021MS002941, 2022.

[26] Elizabeth A Barnes, Benjamin Toms, James W Hurrell, Imme Ebert-Uphoff, Chuck Anderson, and David Anderson. Indicator patterns of forced change learned by an artificial neural network. *Journal of Advances in Modeling Earth Systems*, 12(9):e2020MS002195, 2020.

[27] Charles Anderson and Jason Stock. An interpretable model of climate change using correlative learning. *In NeurIPS 2022 Workshop on Tackling Climate Change with Machine Learning*, 2022.

[28] Elizabeth A Barnes, James W Hurrell, Imme Ebert-Uphoff, Chuck Anderson, and David Anderson. Viewing forced climate patterns through an ai lens. *Geophysical Research Letters*, 46(22):13389–13398, 2019.

[29] Oliver Watt-Meyer, Gideon Dresdner, Jeremy McGibbon, Spencer K Clark, Brian Henn, James Duncan, Noah D Brenowitz, Karthik Kashinath, Michael S Pritchard, Boris Bonev, et al. Ace: A fast, skillful learned global atmospheric model for climate prediction. *arXiv preprint arXiv:2310.02074*, 2023.

[30] Jonathan A Weyn, Dale R Durran, and Rich Caruana. Can machines learn to predict weather? using deep learning to predict gridded 500-hpa geopotential height from historical weather data. *Journal of Advances in Modeling Earth Systems*, 11(8):2680–2693, 2019.

[31] Charles Anderson, Jason Stock, and David Anderson. Interpretable climate change modeling with progressive cascade networks. *arXiv preprint arXiv:2205.06351*, 2022.

[32] Katherine Haynes, Jason Stock, Jack Dostalek, Charles Anderson, and Imme Ebert-Uphoff. Exploring the use of machine learning to improve vertical profiles of temperature and moisture. *Artificial Intelligence for the Earth Systems*, 3(1):e220090, 2024.

[33] Hyeon Kyu Lee and Young-Seok Choi. Application of continuous wavelet transform and convolutional neural network in decoding motor imagery Brain-Computer interface. *Entropy*, 21(12):1199, December 2019.

[34] Hui Liu, Xi-Wei Mi, and Yan-Fei Li. Wind speed forecasting method based on deep learning strategy using empirical wavelet transform, long short term memory neural network and elman neural network. *Energy Convers. Manage.*, 156:498–514, January 2018.

[35] Ryan Lagerquist, Amy McGovern, Cameron R Homeyer, David John Gagne, II, and Travis Smith. Deep learning on Three-Dimensional multiscale data for Next-Hour tornado prediction. *Mon. Weather Rev.*, 148(7):2837–2861, June 2020.

[36] John L Cintineo, Michael J Pavolonis, Justin M Sieglaff, Anthony Wimmers, Jason Brunner, and Willard Bellon. A Deep-Learning model for automated detection of intense midlatitude convection using geostationary satellite images. *Weather Forecast.*, 35(6):2567–2588, December 2020.

[37] Buo-Fu Chen, Boyo Chen, Hsuan-Tien Lin, and Russell L Elsberry. Estimating tropical cyclone intensity by satellite imagery utilizing convolutional neural networks. *Weather Forecast.*, 34(2): 447–465, April 2019.

[38] Mirco Ravanelli and Yoshua Bengio. Speaker recognition from raw waveform with SincNet. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 1021–1028, December 2018.

[39] Maria Ximena Bastidas Rodriguez, Adrien Gruson, Luisa Polania, Shin Fujieda, Flavio Prieto, Kohei Takayama, and Toshiya Hachisuka. Deep adaptive wavelet network. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3111–3119, 2020.

[40] Stéphane Mallat. Group invariant scattering. *Commun. Pure Appl. Math.*, 65(10):1331–1398, October 2012.

[41] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1872–1886, August 2013.

[42] L Sifre and S Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. *Proceedings of the IEEE conference on*, 2013.

[43] Edouard Oyallon, Sergey Zagoruyko, Gabriel Huang, Nikos Komodakis, Simon Lacoste-Julien, Matthew Blaschko, and Eugene Belilovsky. Scattering networks for hybrid representation learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(9):2208–2221, September 2019.

[44] Lennart Bargsten, Katharina A Riedl, Tobias Wissel, Fabian J Brunner, Klaus Schaefers, Michael Grass, Stefan Blankenberg, Moritz Seiffert, and Alexander Schlaefer. Attention via scattering transforms for segmentation of small intravascular ultrasound data sets. In Mattias Heinrich, Qi Dou, Marleen de Bruijne, Jan Lellmann, Alexander Schläfer, and Floris Ernst, editors, *Proceedings of the Fourth Conference on Medical Imaging with Deep Learning*, volume 143 of *Proceedings of Machine Learning Research*, pages 34–47. PMLR, 2021.

[45] Stéphane Mallat. Understanding deep convolutional networks. *Philos. Trans. A Math. Phys. Eng. Sci.*, 374(2065):20150203, April 2016.

[46] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

[47] Xander Davies, Lauro Langosco, and David Krueger. Unifying grokking and double descent. *arXiv preprint arXiv:2303.06173*, 2023.

[48] Steven D Miller, William C Straka, 3rd, Jia Yue, Steven M Smith, M Joan Alexander, Lars Hoffmann, Martin Setvák, and Philip T Partain. Upper atmospheric gravity wave details revealed in nightglow satellite imagery. *Proc. Natl. Acad. Sci. U. S. A.*, 112(49):E6728–35, December 2015.

[49] Tim Schmit, Mat Gunshor, Gang Fu, Tom Rink, Kaba Bah, and Walter Wolf. Goes-r advanced baseline imager (abi) algorithm theoretical basis document for: Cloud and moisture imagery product (cmip), version 2.3. *NOAA NESDIS STAR Document*, page 66, 2010.

[50] Chull Hwan Song, Hye Joo Han, and Yannis Avrithis. All the attention you need: Global-local, spatial-channel attention for image retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2754–2763, 2022.

[51] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.

[52] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[53] Yiliang Zeng, Christian Ritz, Jiahong Zhao, and Jinhui Lan. Attention-Based residual network with scattering transform features for hyperspectral unmixing with limited training samples. *Remote Sensing*, 12(3):400, January 2020.

[54] Mathieu Andreux, Tomás Angles, Georgios Exarchakis, Roberto Leonarduzzi, Gaspar Rochette, Louis Thiry, John Zarka, Stéphane Mallat, Joakim Andén, Eugene Belilovsky, Joan Bruna, Vincent Lostanlen,

Muawiz Chaudhary, Matthew J. Hirn, Edouard Oyallon, Sixin Zhang, Carmine Cella, and Michael Eickenberg. Kymatio: Scattering transforms in python. *Journal of Machine Learning Research*, 21 (60):1–6, 2020. URL http://jmlr.org/papers/v21/19-047.html.

[55] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-Excitation networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(8):2011–2023, August 2020.

[56] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned contextual feature reweighting for image geo-localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2136–2145, 2017.

[57] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[58] John P Cangialosi, Eric Blake, Mark DeMaria, Andrew Penny, Andrew Latto, Edward Rappaport, and Vijay Tallapragada. Recent progress in tropical cyclone intensity forecasting at the national hurricane center. *Weather and Forecasting*, 35(5):1913–1922, 2020.

[59] Manil Maskey, Rahul Ramachandran, Iksha Gurung, Brian Freitag, Muthukumaran Ramasubramanian, and Jeffrey Miller. Tropical cyclone wind estimation competition dataset. https://doi.org/10.34911/rdnt.xs53up, 2018. Version 1.0, Radiant MLHub; Accessed 13 June 2022.

[60] Igor Ivanov. Winners of the wind-dependent variables: Predict wind speeds of tropical storms competition. https://github.com/drivendataorg/wind-dependent-variables/blob/main/1st_Place/reports/DrivenData-Competition-Winner-Documentation.pdf, 2021.

[61] David John Gagne, II, Gunther Wallach, Charlie Becker, and Bill Petzke. Ai4ess summer school hackathon 2020. https://github.com/NCAR/ai4ess-hackathon-2020, 2020.

[62] Steven J Goodman, D Mach, WJ Koshak, and RJ Blakeslee. Glm lightning cluster-filter algorithm (lcfa) algorithm theoretical basis document (atbd). *NOAA NESDIS STAR Document*, page 77, 2010.

[63] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR, 2017.

[64] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):185–207, 2012.

[65] Kristin Koch, Judith McLean, Ronen Segev, Michael A Freed, Michael J Berry II, Vijay Balasubramanian, and Peter Sterling. How much the eye tells the brain. *Current biology*, 16(14):1428–1434, 2006.

[66] Anne M Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136, 1980.

[67] Jan Theeuwes. Top–down and bottom–up control of visual selection. *Acta psychologica*, 135(2): 77–99, 2010.

[68] Michel Failing and Jan Theeuwes. Selection history: How reward modulates selectivity of visual attention. *Psychonomic bulletin & review*, 25(2):514–538, 2018.

[69] Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature reviews neuroscience*, 2(3):194–203, 2001.

[70] Edward Awh, Artem V Belopolsky, and Jan Theeuwes. Top-down versus bottom-up attentional control: A failed theoretical dichotomy. *Trends in cognitive sciences*, 16(8):437–443, 2012.

[71] Eileen Kowler. Eye movements: The past 25 years. *Vision research*, 51(13):1457–1483, 2011.

[72] Min Zhao, Timothy M Gersch, Brian S Schnitzer, Barbara A Dosher, and Eileen Kowler. Eye movements and attention: The role of pre-saccadic shifts of attention in perception, memory and the control of saccades. *Vision research*, 74:40–60, 2012.

[73] Christof Koch and Shimon Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of intelligence*, pages 115–141. Springer, 1987.

[74] Steven P Tipper, Sarah Grison, and Klaus Kessler. Long-term inhibition of return of attention. *Psychological Science*, 14(1):19–25, 2003.

[75] Ernst Niebur and Christof Koch. Control of selective visual attention: Modeling the "where" pathway. *Advances in neural information processing systems*, 8, 1995.

[76] Nicolas Riche, Matthieu Duvinage, Matei Mancas, Bernard Gosselin, and Thierry Dutoit. Saliency and human fixations: State-of-the-art and study of comparison metrics. In *Proceedings of the IEEE international conference on computer vision*, pages 1153–1160, 2013.

[77] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[78] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[79] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.

[80] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018.

[81] Debang Li, Junge Zhang, and Kaiqi Huang. Learning to learn cropping models for different aspect ratio requirements. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12685–12694, 2020.

[82] Wenguan Wang, Jianbing Shen, and Haibin Ling. A deep network solution for attention and aesthetics aware photo cropping. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1531–1544, 2018.

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

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

[85] Hugo Larochelle and Geoffrey E Hinton. Learning to combine foveal glimpses with a third-order boltzmann machine. *Advances in neural information processing systems*, 23, 2010.

[86] Volodymyr Mnih, Nicolas Heess, and Alex Graves. Recurrent models of visual attention. *Advances in neural information processing systems*, 27, 2014.

[87] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*, 2014.

[88] Sean Welleck, Jialin Mao, Kyunghyun Cho, and Zheng Zhang. Saliency-based sequential image attention with multiset prediction. *Advances in neural information processing systems*, 30, 2017.

[89] Gamaleldin Elsayed, Simon Kornblith, and Quoc V Le. Saccader: improving accuracy of hard attention models for vision. *Advances in Neural Information Processing Systems*, 32, 2019.

[90] Sweta Kumari and V Srinivasa Chakravarthy. Biologically inspired image classifier based on saccadic eye movement design for convolutional neural networks. *Neurocomputing*, 513:294–317, 2022.

[91] Leo Schwinn, Doina Precup, Bjoern Eskofier, and Dario Zanca. Simulating human gaze with neural visual attention. In *NeuRIPS 2022 Workshop on Gaze Meets ML*, 2022.

[92] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[93] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

[94] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

[95] Lingchen Meng, Hengduo Li, Bor-Chun Chen, Shiyi Lan, Zuxuan Wu, Yu-Gang Jiang, and Ser-Nam Lim. Adavit: Adaptive vision transformers for efficient image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12309–12318, 2022.

[96] Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. Evo-vit: Slow-fast token evolution for dynamic vision transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2964–2972, 2022.

[97] Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10809–10818, 2022.

[98] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022.

[99] Andreas Fidjeland. Cluttered mnist dataset. https://github.com/deepmind/mnist-cluttered, 2015.

[100] Nicola Maher, Sebastian Milinski, and Ralf Ludwig. Large ensemble climate model simulations: introduction, overview, and future prospects for utilising multiple types of large ensemble. *Earth System Dynamics*, 12(2):401–418, 2021.

[101] Justin S Mankin, Flavio Lehner, Sloan Coats, and Karen A McKinnon. The value of initial condition large ensembles to robust adaptation decision-making. *Earth's Future*, 8(10):e2012EF001610, 2020.

[102] Benjamin M Sanderson, Keith W Oleson, Warren G Strand, Flavio Lehner, and Brian C O'Neill. A new ensemble of gcm simulations to assess avoided impacts in a climate mitigation scenario. *Climatic Change*, 146:303–318, 2018.

[103] Christopher B Field and Vicente R Barros. *Climate change 2014–Impacts, adaptation and vulnerability: Regional aspects*. Cambridge University Press, 2014.

[104] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[105] Benjamin A Toms, Elizabeth A Barnes, and Imme Ebert-Uphoff. Physically interpretable neural networks for the geosciences: Applications to earth system variability. *Journal of Advances in Modeling Earth Systems*, 12(9):e2019MS002002, 2020.

[106] Veronika Eyring, Sandrine Bony, Gerald A Meehl, Catherine A Senior, Bjorn Stevens, Ronald J Stouffer, and Karl E Taylor. Overview of the coupled model intercomparison project phase 6 (cmip6) experimental design and organization. *Geoscientific Model Development*, 9(5):1937–1958, 2016.

[107] Donald W Marquardt and Ronald D Snee. Ridge regression in practice. *The American Statistician*, 29 (1):3–20, 1975.

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

[109] Thomas A Jones, Patrick Skinner, Nusrat Yussouf, Kent Knopfmeier, Anthony Reinhart, Xuguang Wang, Kristopher Bedka, William Smith Jr, and Rabindra Palikonda. Assimilation of goes-16 radiances and retrievals into the warn-on-forecast system. *Monthly Weather Review*, 148(5):1829–1859, 2020.

[110] William E Line, Timothy J Schmit, Daniel T Lindsey, and Steven J Goodman. Use of geostationary super rapid scan satellite imagery by the storm prediction center. *Weather and Forecasting*, 31(2): 483–494, 2016.

[111] Travis M Smith, Valliappa Lakshmanan, Gregory J Stumpf, Kiel L Ortega, Kurt Hondl, Karen Cooper, Kristin M Calhoun, Darrel M Kingfield, Kevin L Manross, Robert Toomey, et al. Multi-radar multi-sensor (mrms) severe weather and aviation products: Initial operating capabilities. *Bulletin of the American Meteorological Society*, 97(9):1617–1630, 2016.

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

[113] Mingming Zhu, Qi Liao, Lin Wu, Si Zhang, Zifa Wang, Xiaole Pan, Qizhong Wu, Yangang Wang, and Debin Su. Multiscale representation of radar echo data retrieved through deep learning from numerical model simulations and satellite images. *Remote Sensing*, 15(14):3466, 2023.

[114] Kyle A. Hilburn, Imme Ebert-Uphoff, and Steven D. Miller. Development and interpretation of a neural-network-based synthetic radar reflectivity estimator using goes-r satellite observations. *Journal of Ap-*

*plied Meteorology and Climatology*, 60(1):3–21, 2021. doi: https://doi.org/10.1175/JAMC-D-20-0084.1. URL https://journals.ametsoc.org/view/journals/apme/60/1/jamc-d-20-0084.1.xml.

[115] Mingshan Duan, Jiangjiang Xia, Zhongwei Yan, Lei Han, Lejian Zhang, Hanmeng Xia, and Shuang Yu. Reconstruction of the radar reflectivity of convective storms based on deep learning and himawari-8 observations. *Remote Sensing*, 13(16):3330, 2021.

[116] Xiaoqi Yu, Xiao Lou, Yan Yan, Zhongwei Yan, Wencong Cheng, Zhibin Wang, Deming Zhao, and Jiangjiang Xia. Radar echo reconstruction in oceanic area via deep learning of satellite data. *Remote Sensing*, 15(12):3065, 2023.

[117] Kyle Hilburn. Gremlin conus3 dataset for 2020, 2023. Dryad, Dataset, https://doi.org/10.5061/dryad.h9w0vt4nq.

[118] Kyle Hilburn. Gremlin conus3 dataset for 2021, 2023. Dryad, Dataset, https://doi.org/10.5061/dryad.zs7h44jf2.

[119] Kyle Hilburn. Gremlin conus3 dataset for 2022, 2023. Dryad, Dataset, https://doi.org/10.5061/dryad.2jm63xstt.

[120] Steven J Goodman, Richard J Blakeslee, William J Koshak, Douglas Mach, Jeffrey Bailey, Dennis Buechler, Larry Carey, Chris Schultz, Monte Bateman, Eugene McCaul Jr, et al. The goes-r geostationary lightning mapper (glm). *Atmospheric research*, 125:34–49, 2013.

[121] Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. Revealing the dark secrets of bert. *arXiv preprint arXiv:1908.08593*, 2019.

[122] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*, 2019.

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

[124] Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. On identifiability in transformers. *arXiv preprint arXiv:1908.04211*, 2019.

[125] Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. *arXiv preprint arXiv:1908.04626*, 2019.

[126] Sofia Serrano and Noah A Smith. Is attention interpretable? *arXiv preprint arXiv:1906.03731*, 2019.

[127] Yaru Hao, Li Dong, Furu Wei, and Ke Xu. Self-attention attribution: Interpreting information interactions inside transformer. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence*. AAAI Press, 2021. URL https://arxiv.org/pdf/2004.11207.pdf.

[128] Kedar Dhamdhere, Mukund Sundararajan, and Qiqi Yan. How important is a neuron? *arXiv preprint arXiv:1805.12233*, 2018.

[129] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 782–791, 2021.

[130] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, volume 139, pages 10347–10357, July 2021.

[131] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[132] Andrew S. Latto and John P. Cangialosi. National hurricane center tropical cyclone report: Tropical storm colin. *Miami: National Hurricane Center*, 2022.

[133] William KM Lau, Duane E Waliser, and Duane Waliser. *Predictability and forecasting*. Springer, 2005.

[134] Morteza Mardani, Noah Brenowitz, Yair Cohen, Jaideep Pathak, Chieh-Yu Chen, Cheng-Chin Liu, Arash Vahdat, Karthik Kashinath, Jan Kautz, and Mike Pritchard. Residual diffusion modeling for km-scale atmospheric downscaling. *arXiv preprint arXiv:2309.15214*, 2024.

[135] Lizao Li, Rob Carver, Ignacio Lopez-Gomez, Fei Sha, and John Anderson. Seeds: Emulation of weather forecast ensembles with diffusion models. *arXiv preprint arXiv:2306.14066*, 2023.

[136] Pritthijit Nath, Pancham Shukla, and César Quilodrán-Casas. Forecasting tropical cyclones with cascaded diffusion models. *arXiv preprint arXiv:2310.01690*, 2023.

[137] Zhihan Gao, Xingjian Shi, Boran Han, Hao Wang, Xiaoyong Jin, Danielle Maddix, Yi Zhu, Mu Li, and Yuyang Wang. Prediff: Precipitation nowcasting with latent diffusion models. *arXiv preprint arXiv:2307.10422*, 2023.

[138] Jussi Leinonen, Ulrich Hamann, Daniele Nerini, Urs Germann, and Gabriele Franch. Latent diffusion models for generative precipitation nowcasting with accurate uncertainty quantification. *arXiv preprint arXiv:2304.12891*, 2023.

[139] Christian Kummerow, William Barnes, Toshiaki Kozu, James Shiue, and Joanne Simpson. The tropical rainfall measuring mission (trmm) sensor package. *Journal of atmospheric and oceanic technology*, 15(3):809–817, 1998.

[140] Arthur Y Hou, Ramesh K Kakar, Steven Neeck, Ardeshir A Azarbarzin, Christian D Kummerow, Masahiro Kojima, Riko Oki, Kenji Nakamura, and Toshio Iguchi. The global precipitation measurement mission. *Bulletin of the American meteorological Society*, 95(5):701–722, 2014.

[141] N Chen, A J Majda, and D Giannakis. Predicting the cloud patterns of the Madden-Julian oscillation through a low-order nonlinear stochastic model. *Geophys. Res. Lett.*, August 2014. ISSN 0094-8276, 1944-8007. doi: 10.1002/2014GL060876.

[142] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[143] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

[144] Giorgio Parisi. Correlation functions and computer simulations. *Nuclear Physics B*, 180(3):378–384, 1981.

[145] Ulf Grenander and Michael I Miller. Representations of knowledge in complex systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(4):549–581, 1994.

[146] G.J. Huffman, E.F. Stocker, D.T. Bolvin, E.J. Nelkin, and Jackson Tan. Gpm imerg final precipitation l3 half hourly 0.1 degree x 0.1 degree v06. Greenbelt, MD, Goddard Earth Sciences Data and Information Services Center (GES DISC), Accessed: March 19, 2022., 2019. 10.5067/GPM/IMERG/3B-HH/06.

[147] George J Huffman, David T Bolvin, Dan Braithwaite, Kuolin Hsu, Robert Joyce, Pingping Xie, and Soo-Hyun Yoo. Nasa global precipitation measurement (gpm) integrated multi-satellite retrievals for gpm (imerg). *Algorithm theoretical basis document (ATBD) version*, 4(26):30, 2015.

[148] Georgy Ayzel, Tobias Scheffer, and Maik Heistermann. Rainnet v1. 0: a convolutional neural network for radar-based precipitation nowcasting. *Geoscientific Model Development*, 13(6):2631–2644, 2020.

[149] Ernest Hovmöller. The trough-and-ridge diagram. *Tellus*, 1(2):62–66, 1949.

[150] Roland A Madden and Paul R Julian. Observations of the 40–50-day tropical oscillation—a review. *Monthly weather review*, 122(5):814–837, 1994.

[151] Putian Zhou, Lingling Suo, Jiacan Yuan, and Benkui Tan. The east pacific wavetrain: Its variability and impact on the atmospheric circulation in the boreal winter. *Advances in Atmospheric Sciences*, 29: 471–483, 2012.

[152] Matthew Wheeler and George N Kiladis. Convectively coupled equatorial waves: Analysis of clouds and temperature in the wavenumber–frequency domain. *Journal of the Atmospheric Sciences*, 56(3): 374–399, 1999.

[153] Michael Maier-Gerber. A python package for the construction of the wheeler–kiladis space-time spectra. https://github.com/mmaiergerber/wk_spectra, 2018.

[154] Michio Yanai and Takio Maruyama. Stratospheric wave disturbances propagating over the equatorial pacific. *Journal of the Meteorological Society of Japan. Ser. II*, 44(5):291–294, 1966.

[155] Katherine H Straub and George N Kiladis. Observations of a convectively coupled kelvin wave in the eastern pacific itcz. *Journal of the Atmospheric Sciences*, 59(1):30–53, 2002.

[156] George N Kiladis, Katherine H Straub, and Patrick T Haertel. Zonal and vertical structure of the madden–julian oscillation. *Journal of the atmospheric sciences*, 62(8):2790–2809, 2005.

[157] Nigel M Roberts and Humphrey W Lean. Scale-selective verification of rainfall accumulations from high-resolution forecasts of convective events. *Monthly Weather Review*, 136(1):78–97, 2008.

[158] Nigel Roberts. Assessing the spatial and temporal variation in the skill of precipitation forecasts from an nwp model. *Meteorological Applications: A journal of forecasting, practical applications, training techniques and modelling*, 15(1):163–169, 2008.

[159] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, et al. The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020.

[160] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

[161] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pages 13213–13232. PMLR, 2023.

[162] Jason Stock. Using machine learning to improve vertical profiles of temperature and moisture for severe weather nowcasting. Master's thesis, Colorado State University, 2021.

[163] Elliott M Forney. *Convolutional Neural Networks for EEG Signal Classification in Asynchronous Brain-Computer Interfaces*. Colorado State University, 2019.

[164] Christopher Velden, Bruce Harper, Frank Wells, John L Beven, Ray Zehr, Timothy Olander, Max Mayfield, Charles "CHIP" Guard, Mark Lander, Roger Edson, et al. The dvorak tropical cyclone intensity estimation technique: A satellite-based method that has endured for over 30 years. *Bulletin of the American Meteorological Society*, 87(9):1195–1210, 2006.

[165] Karthik Balaguru, Chuan-Chieh Chang, L Ruby Leung, Gregory R Foltz, Samson M Hagos, Michael F Wehner, James P Kossin, Mingfang Ting, and Wenwei Xu. A global increase in nearshore tropical cyclone intensification. *Earth's Future*, 12(5):e2023EF004230, 2024.

[166] Albenis Pérez-Alarcón, José C Fernández-Alvarez, and Patricia Coll-Hidalgo. Global increase of the intensity of tropical cyclones under global warming based on their maximum potential intensity and cmip6 models. *Environmental Processes*, 10(2):36, 2023.

[167] David Manning, Susan Ethell, Tim Donovan, and Trevor Crawford. How do radiologists do it? the influence of experience and training on searching for chest nodules. *Radiography*, 12(2):134–142, 2006.

[168] Laura McLaughlin, Raymond Bond, Ciara Hughes, Jonathan McConnell, and Sonyia McFadden. Computing eye gaze metrics for the automatic assessment of radiographer performance during x-ray image interpretation. *International journal of medical informatics*, 105:11–21, 2017.

[169] Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. *URL https://openaipublic. blob. core. windows. net/neuron-explainer/paper/index. html.(Date accessed: 14.05. 2023)*, 2, 2023.

[170] Annie PS Wong, Susan E Wijffels, Stephen C Riser, Sylvie Pouliquen, Shigeki Hosoda, Dean Roemmich, John Gilson, Gregory C Johnson, Kim Martini, David J Murphy, et al. Argo data 1999–2019: Two million temperature-salinity profiles and subsurface velocity observations from a global array of profiling floats. *Frontiers in Marine Science*, 7:700, 2020.