A Deep Learning Investigation of Zonal Jets

by

© **Nurul Bin Ibrahim**

A thesis submitted to the Department of Physics and Physical Oceanography in partial fulfillment of the requirements for the degree of Bachelor of Science (Honours).

Department of Physics and Physical Oceanography
Memorial University of Newfoundland

May 2024

St. John's, Newfoundland and Labrador, Canada

Abstract

Zonal jets play a crucial role in influencing the dynamics and transport properties of planetary atmospheres and oceans. Despite their importance, the precise mechanisms governing their formation, equilibration, and maintenance remain a topic of ongoing investigation. This thesis aims to leverage deep learning techniques to gain new insights about zonal jets and develop efficient predictive models. We focus on applying these techniques to a low-order model (LOM) of the stochastically excited two-dimensional Boussinesq system, which can capture the essential interactions between the zonal jet and the associated turbulence. Various deep learning architectures like feed-forward neural networks, statistics-informed neural networks, and physics-informed neural networks are developed and trained on large ensembles of LOM simulations to analyze the dynamics of zonal jets. We demonstrate that such deep learning models can capture the underlying statistical properties and predict the long-term behavior of these jets. The capabilities of the trained models are also explored by extracting physical insights based on the analysis. For example, we can use our neural network to understand the physical mechanism behind jet maintenance. These findings have implications for improving our understanding and ability to predict geophysical turbulence. We also compare our deep learning results with the results of the statistical state dynamics (SSD) theory of turbulence. This comparison is used to test the correspondence of SSD with the real behaviour of the stochastic system and identify areas where it could be improved.

To my parents
whose love and guidance created a nurturing home for me,
and to my Physics Department
where I was welcomed into a second home.

Acknowledgements

Reflecting on the journey that has brought me to this moment, I'm filled with a deep sense of gratitude for the people who have made an immeasurable impact on my life. The constant support, guidance, and inspiration from a great network of mentors, colleagues, friends, family, and everyone else I am involved with, have been instrumental in shaping this thesis, making it a reflection of their contributions as much as my own work.

First and foremost, I want to express my heartfelt thanks to my incredible supervisors, Dr. Joseph G. Fitzgerald and Dr. Terrence Tricco. Their unwavering support, valuable guidance, and genuine interest in my growth have been a constant source of inspiration. They've gone above and beyond, not just academically, but also in helping me develop as a person. Their confidence in me has been a driving force, encouraging me to push beyond my limits and explore new opportunities. From sending me to conferences across Canada to introducing me to exciting projects, they've opened doors and ignited a passion within me that will shape my future endeavors.

I can't go without mentioning the immense support of Dr. Len Zedel, the Head of our Physics and Physical Oceanography department. He was the one who first introduced me to the fascinating world of scientific research and the field of Physical Oceanography. Dr. Zedel was committed to ensuring I had the most rewarding and enjoyable experience in a domain that was once new to me. This has been truly remarkable. The foundation he helped build has been crucial in shaping me into the passionate scientist I am today.

The Physics department has been more than just an academic setting – it has been a home filled with people who have welcomed me warmly. From my classmates to the staff and faculty, the love, encouragement, and support I have received have been truly amazing. As an international student, coming from far away, the feeling of belonging and acceptance I have found here shows how great the people in this department are. I want to sincerely thank Donna, Lisa, Chris, Rick, Melissa, and Michelle, who have treated me like a son and a friend, making my experience even more special.

My family's role in this journey cannot be overstated. My parents have been my ultimate support system, always believing in me and encouraging me to pursue my dreams, even when they seemed out of reach. Their sacrifices and unconditional love have been the driving force behind my success, and I am forever grateful for the values they have instilled in me. Besides my parents, my extended family members, those who I grew up with, and those who raised me to be resilient and compassionate, have also played a crucial role. Their collective wisdom and the sense of community they fostered have provided a strong foundation upon which I have built my aspirations and achievements. Their relentless support and encouragement have been indispensable, reminding me of the importance of family and roots in one's life journey.

I must also acknowledge the invaluable role of my friends, who have accompanied me throughout this journey, from both close by and afar. The friends from my high school days, with whom I began this exciting adventure in a new country, and those I met in college, who welcomed me into their circle, have consistently provided support and friendship. The cherished memories we have crafted together and the strong bonds we have developed are treasures that I will carry with me forever. Among the many who have supported me, a special thanks goes to those who have been consistently supportive during the most demanding phases of this work.

I am overwhelmed with gratitude for all the people who have motivated me, supported me, and believed in me throughout this incredible journey. Each and every person mentioned here, and countless others who have touched my life, have played an integral role in shaping the scientist, the individual, and the dreamer that I am today. As I close this chapter and move forward, I carry with me the lessons, the love, and the inspiration that you've all so generously bestowed upon me. From the bottom of my heart, thank you.

Table of contents

Chapter 1

Introduction

1.1 General Topic and Background

Zonal jets are prominent features in the Earth's atmosphere and ocean, as well as in the atmospheres of other planets in the solar system. They are characterized by strong, east-west oriented currents and often play crucial roles in shaping planetary climates and circulation patterns [28]. The velocities of these jets vary across latitudes or depths, creating distinct regions with different flow characteristics [28, 69, 18, 6, 2].

On Earth, zonal jets in the atmosphere such as the polar jet stream play a crucial role in the general circulation of the climate system, governing the transport of heat and moisture and affecting the dynamics of weather and climate [32, 74]. In the oceans, jets such as the Antarctic Circumpolar Current (ACC) are key drivers of the global thermohaline circulation, influencing the distribution of heat, salt, and nutrients, which in turn affect climate patterns. The Equatorial Deep Jets (EDJs) are important for the transport of oxygen in the equatorial ocean, and it has been speculated that they also drive atmospheric variability on certain timescales [52, 9, 79, 14].

Beyond Earth, zonal jets are also a defining characteristic of the gas giants in our solar system, particularly Jupiter and Saturn. The distinct banded patterns visible on these planets are associated with powerful zonal jets, which provide invaluable observational examples of atmospheric climates where jets are a dominant feature. The existence of zonal jets on both Earth and other planets suggests that comprehensive theories of zonal jet formation and maintenance should be capable of explaining their occurrence and characteristics across a wide range of planetary environments. [79, 82].

1.2 Motivation

The study of zonal jets is essential for understanding atmospheric and oceanic dynamics as they play a crucial role in influencing large-scale circulation and tracer transport [6]. Zonal jet systems also provide important examples of the interactions between turbulence and coherent flows. As interactions of this type are ubiquitous in the climate system, studies of zonal jets present valuable opportunities to advance our understanding of these dynamics.[28, 8].

Interactions between turbulence and the mean flow influence the dynamics of both weather and climate. Weather and climate models rely on simplified models called ‘parameterizations’ to represent these interactions. Studies of zonal jets are critical to further inform our understanding of these interactions, which underpins the development of improved parameterizations. Zonal jets significantly influence atmospheric and oceanic phenomena, affecting precipitation distribution, extreme weather event frequency, and long-term climate patterns [1, 49]. Improved representation of these jets and their associated interactions in predictive models can lead to more accurate forecasts of atmospheric and oceanic states, benefiting sectors reliant on weather and climate forecasts, such as agriculture, water management, and emergency response planning [3, 49].

Zonal jets present an opportunity to deepen our fundamental understanding of fluid dynamics and turbulence. These jets arise from and are maintained by a balance of forces, including planetary rotation, stratification, and turbulence. Their study offers a window into understanding physical processes such as inverse energy cascades, wave mean flow interactions, potential vorticity mixing, and the dynamics of the statistical state of the turbulence [62, 77, 20, 24]. Insights gained from zonal jets improve our understanding of the principles that govern rotating, stratified fluids, with broader applications across astrophysics and plasma physics.

1.3 Study of Zonal Jets

Observational studies have been instrumental in characterizing zonal jets in the atmosphere and oceans of the Earth. Techniques such as satellite measurements and in situ data collection have provided valuable insight into the properties of these jets, including their strength, structure, and temporal variations [52, 22]. These studies have highlighted the widespread presence of zonal jets in the global ocean and their essential function in the transport of heat and nutrients [75]. Atmospheric observations have further elucidated the variability of these jets, their impact on weather patterns, and their sensitivity to climate change.

To understand the origins and behaviors of zonal jets, researchers have developed various theoretical models. These models range from conceptual frameworks like the Rhines scale and potential vorticity staircase to more comprehensive mathematical theories such as statistical state dynamics (SSD), also known as stochastic structural stability theory (SSST) or the second-order cumulant expansion (CE2) [27, 62, 43, 20, 25]. These models have emphasized the significance of inverse energy cascade, potential vorticity mixing, and eddy-mean flow interactions in jet formation and maintenance. While these models offer valuable insights, they also rely on major simplifications of fluid dynamics, such as the quasi-linear approximation and diffusion of energy in spectral space. These simplifications are never exactly realized in fundamental fluid dynamics, so data-driven methods based on observations, laboratory experiments, or direct numerical simulations are desirable to examine the accuracy of these models.

Numerical simulations have become essential for investigating zonal jet dynamics due to the lack of analytical solutions to the nonlinear equations of fluid dynamics that represent turbulent flows. These simulations have been instrumental in dissecting the features of zonal jets, exploring their interactions with eddies and waves, and understanding their responses to different forcing and dissipation scenarios [8, 39, 4]. Simulations have also been used to study conceptual predictions such as the anisotropic inverse energy cascade, and the Rhines scale [72]. However, the computational resources required by these simulations, especially at high Reynolds numbers, constrain the scope of parameter exploration and the time duration of the numerical integrations.

1.4 Traditional Challenges

The study of zonal jets presents a significant challenge due to the multiscale nature of their dynamics. These jets arise from interactions between large-scale mean flows and smaller-scale turbulent eddies, spanning a range of spatial and temporal scales [77]. High-resolution simulations are essential for capturing this range of scales, which in turn requires significant computational resources. The nonlinear and chaotic characteristics of turbulent flows add another layer of complexity, making it challenging to isolate and comprehend the various physical processes governing the formation and persistence of zonal jets.

One of the main obstacles to studying zonal jets is the scarcity of observational data, particularly in deep-ocean regions [85]. Although satellite observations and in situ measurements have provided insight into the structure and variability of zonal jets in the Earth's atmosphere and upper ocean, deep-ocean data is less comprehensive. [74]. Observations in the deep ocean rely on a sparse network of moorings and hydrographic surveys, making it difficult to validate theoretical models and numerical simulations [5].

1.5 Deep Learning of Zonal Jet Dynamics

Deep learning, a subset of machine learning, has been instrumental in addressing a wide variety of challenges in many scientific fields. Deep neural networks excel in identifying important patterns within large datasets and often outperform the capabilities of conventional methodologies on pattern recognition tasks [46, 30]. In the realm of physical sciences, the versatility of deep learning is evident through its applications ranging from quantum mechanics to physical oceanography [16, 7]. These successes underscore deep learning's potential to enhance our understanding of complex phenomena.

Training deep neural networks on datasets derived from simulations or observations of zonal jet systems has the potential to identify fundamental features and processes underpinning zonal jet formation, equilibration, and maintenance, which are difficult to obtain using traditional methods of analysis. [15, 84].

By assimilating patterns from data, deep neural networks can model the core dynamics of zonal jets with greater computational efficiency than comprehensive simulations [34]. These models can serve various purposes, including predicting zonal jet evolution under different conditions, pinpointing parameters conducive to jet formation, and evaluating the impact of initial states and model assumptions on outcomes. Furthermore, deep learning facilitates the creation of hybrid models like the physics-informed neural network that can potentially merge data-driven insights with physics-based principles to produce more precise and reliable predictions of zonal jet dynamics [50].

The goal of this thesis is to apply deep learning to the study of zonal jet dynamics. The research questions we aim to tackle are as follows:

1. In what ways can deep learning facilitate identifying the processes that drive zonal jet maintenance?

2. What are the optimal network architectures and training methodologies for deep neural networks modeling zonal jets?

3. How do deep learning-based data-driven models stack up against traditional numerical and theoretical models in terms of precision, computational efficiency, and clarity of interpretation?

4. Is it feasible to use deep learning to develop hybrid models that integrate data-driven and physics-informed approaches, thereby enhancing the accuracy and capability of our models?

1.6 Thesis Structure

This thesis is organized into five main chapters. Chapter **1** introduces the general topic and background, provides a literature review, discusses the need for change and the limitations of traditional methods, and presents the research problem, questions, aims, and objectives. Chapter **2** provides historical context for the study of zonal jets, including their discovery, offering a historical overview of their discovery, observations, and significance to the climates of Earth and other planets. Chapter **3** introduces the theoretical concepts central to this thesis in the areas of zonal jet physics and machine learning, discussing zonal jets and deep learning in detail. Chapter **4** presents the methodology, including an overview of the approach, data generation and exploration, baseline models, and the development of deep learning models such as feed-forward neural networks, statistics-informed neural networks, and physics-informed neural networks. This chapter also covers model training and validation techniques. Chapter **5** presents the results and discussion, interpreting the findings, comparing the performance of different models, and discussing the implications for understanding zonal jet dynamics and the potential for future research.

Chapter 2

History of The Scientific Study of Zonal Jets

The study of zonal jets has a long history, with observations dating back to the 17th century when Giovanni Cassini first observed them on Jupiter [58]. These banded structures can extend for thousands of kilometers and persist for decades or centuries, and have been observed on Jupiter and Saturn, as shown in Figure 2.1 [40].

On Earth, zonal jets were first recognized in the early 20th century with the discovery of the subtropical jet streams and the polar front jet [47]. These atmospheric jets significantly influence weather patterns and climate variability by transporting heat, moisture, and momentum, guiding the large-scale atmospheric disturbances we know as weather systems [37], and acting as waveguides for large-scale atmospheric waves, such as Rossby waves [77]. The position and strength of atmospheric zonal jets affect regional climate patterns, including temperature, precipitation, and the frequency and intensity of extreme weather events [33].

In Earth's oceans, the existence of zonal jets like the Antarctic Circumpolar Current (ACC) and the Equatorial Deep Jets (EDJs) was not recognized until the latter half of the 20th century, with the ACC being discovered through early oceanographic expeditions in the 1950s and 1960s [64], and the EDJs being identified from observational data in the 1970s [26]. The ACC, the strongest current system in the world's oceans, connects the Atlantic, Pacific, and Indian Ocean basins, facilitating the exchange of water masses and the redistribution of heat, salt, and nutrients [64]. The

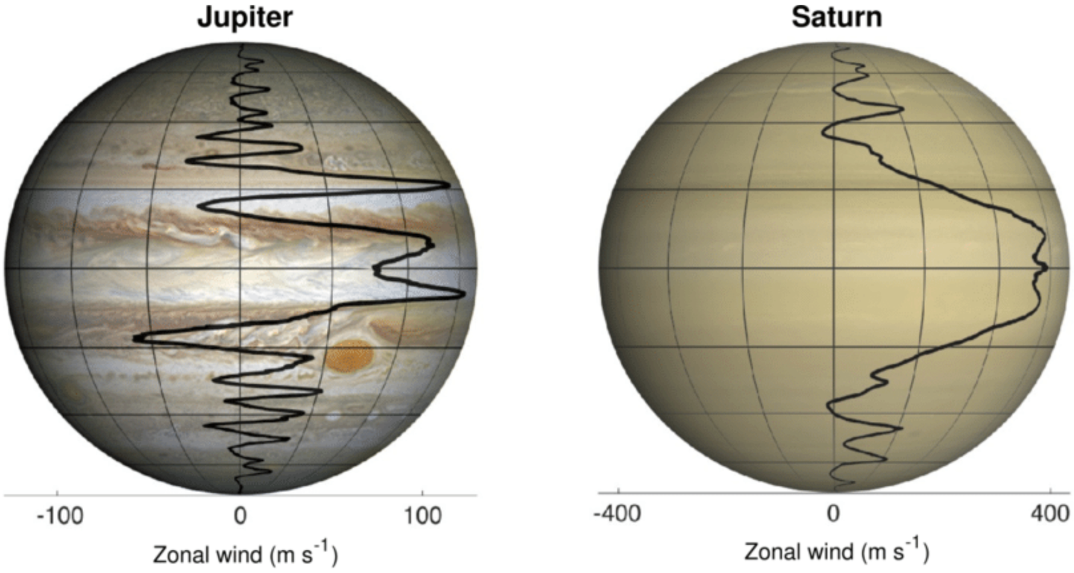

Figure 2.1: Left: Zonal wind profile of Jupiter (Tollefson et al. 2017) superimposed on a photo taken by the Hubble Wide Field Camera in 2014; Right: Zonal wind profile of Saturn superimposed on an image created by Björn Jónsson by combining Cassini and Voyager images and removing the rings.

EDJs, alongside the other components of the equatorial circulation, including the Equatorial Undercurrent (EUC) and the Equatorial Countercurrents (ECCs), are integral to the tropical ocean circulation. These jets transport warm water from the western to the eastern Pacific, influencing El Niño and La Niña events and affecting nutrient distribution and marine ecosystem productivity.

In Figure 2.2, the wavy, ribbon-like patterns in the equatorial regions of the Pacific and Atlantic Oceans indicate the presence and influence of the Equatorial Deep Jets (EDJs) on the surface ocean dynamics and temperatures. While the EDJs themselves are subsurface currents flowing at depths of around 300-500 meters, their signature is evident in the distinct equatorial patterns visible in this visualization of global ocean conditions.

Zonal jets also significantly influence the distribution and migration of marine species, including commercially important fish stocks. The strong currents associated with these jets can act as barriers to the dispersal of marine organisms, creating distinct biogeographic regions. At the same time, the jets can facilitate the long-distance

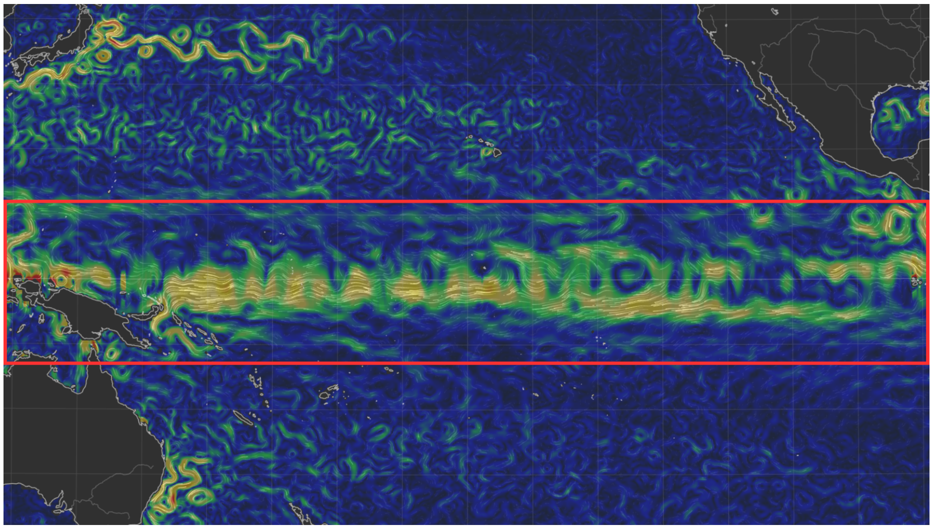

Figure 2.2: Visualization of global ocean currents at the surface, generated using data from the OSCAR (Ocean Surface Current Analyses Real-time) v2.0 dataset and the Earth & Space Research Organization.

transport of larvae and other planktonic organisms, contributing to the connectivity of marine populations across vast distances.

Foundational Studies

The study of zonal jets has been shaped by several seminal works that have shaped the conventional description of their dynamics. One of the most influential contributions was made by Peter B. Rhines in his 1975 paper, "Waves and turbulence on a beta-plane" [62]. In this work, Rhines introduced the concept of what is now known as the "Rhines scale," a characteristic length scale that emerges from the interplay between turbulent eddy interactions and Rossby wave dynamics. The Rhines scale, defined as $L_\beta = \sqrt{U/\beta}$, where U is the root-mean-square velocity, and β is the latitudinal gradient of the Coriolis parameter, sets the width of the zonal jets that form in a turbulent flow constrained by differential rotation (i.e., the beta effect). Rhines' work provided a theoretical framework for understanding the anisotropic cascade of energy in beta-plane turbulence, arguing that energy is preferentially transferred to larger zonal scales until it reaches the Rhines scale, at which point the flow organizes itself into alternating zonal jets.

Numerical simulations have played a crucial role in validating and extending the theoretical predictions of Rhines and others. The pioneering work of Williams (1978) [81] and Vallis and Maltrud (1993) [78] demonstrated the spontaneous emergence of zonal jets in two-dimensional turbulent flows on a beta-plane, again suggesting the importance of the anisotropic inverse energy cascade in the formation of these structures. These studies also highlighted the importance of the Rhines scale in predicting jet structure.

Observational studies have provided compelling evidence for the existence of zonal jets in planetary atmospheres and oceans. The Voyager missions to the outer solar system revealed the stunning banded structure of the giant planets, with zonal wind speeds reaching up to 400 m/s on Jupiter [40]. High-resolution observations of the Earth's atmosphere and oceans have also revealed the presence of multiple zonal jet structures, such as the subtropical jet streams and the equatorial deep jets [52, 63].

Laboratory experiments have further advanced our understanding of zonal jet dynamics. The work of Read et al. (2004) [60] demonstrated the spontaneous emergence of zonal jets in a rotating, stratified fluid in a laboratory setting, providing a valuable experimental analog for the study of planetary atmospheres. Subsequent experiments by Espa et al. (2010) [23] and others have further explored the parameter space of jet formation, investigating the influence of factors such as the strength of the background rotation and the nature of the forcing on the properties of the emergent jets.

Theoretical studies have continued to advance our understanding of the mechanisms governing the formation and maintenance of zonal jets. The application of statistical theories like the SSD has provided new insights into the role of eddy-mean flow interactions in the maintenance of zonal jets [27, 24, 25]. These theories have been successfully applied to geophysical and astrophysical flows, highlighting the universality of the mechanisms underlying zonal jet formation.

Chapter 3

Theoretical Concepts

3.1 Zonal Jets

The formation and maintenance of zonal jets are intimately linked to the presence of turbulent eddies and waves in the flow. The turbulent field of waves and eddies is characterized by its characteristic length scale L and velocity scale U. Turbulence plays a crucial role in the transport of energy, momentum, and tracers in geophysical flows. Understanding the interaction between turbulence and the mean flow is essential for elucidating the mechanisms underlying zonal jet formation and maintenance.

The width of the jets is related to the Rhines scale, $L_R = \sqrt{U/\beta}$, which emerges as a balance between the inverse energy cascade and the beta-effect. The Rhines scale represents the length scale at which the turbulent eddy field transitions from isotropic to anisotropic, and energy begins to be channeled into zonal modes. This scale has been shown to provide a good estimate of the observed width of zonal jets in both numerical simulations and planetary atmospheres [25].

The analogy of zonal jets being a stretched-out version of an eddy provides a useful conceptual framework for understanding the relationship between these two fundamental structures in geophysical turbulence. Consider an eddy with a characteristic length scale L and velocity scale U. If this eddy is stretched in the zonal direction while its meridional extent remains fixed, the resulting structure resembles a zonal jet. The stretching process increases the zonal length scale of the eddy, L_x, while the meridional length scale, L_y, remains unchanged, leading to an increase in the aspect ratio of the structure, defined as L_x/L_y.

To further investigate the dynamics of zonal jets and their interaction with turbulent eddies, it is necessary to delve into the fundamental equations governing fluid motion. The Navier-Stokes equation, which represents the conservation of momentum in a viscous fluid, and the Boussinesq approximation, which simplifies the treatment of stratified flows, provide the foundation for deriving the key equations that describe the evolution of zonal jets.

In the following section, we will present a detailed derivation of the equations that make up the LOM, starting from the Navier-Stokes equation and the Boussinesq approximation. We will explore the physical interpretation of the various terms in the equation and discuss the implications for the dynamics of zonal jets.

Equations & Derivations

To gain a deeper understanding of the dynamics of zonal jet formation and maintenance, we start with the fundamental equations of fluid motion and derive the key equations that describe the evolution of zonal jets and their interaction with turbulent eddies.

The Navier-Stokes equation, which represents the conservation of momentum in a viscous fluid, can be written as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{F}, \tag{3.1}$$

where $\mathbf{u}$ is the velocity vector, t is time, ρ is the fluid density, p is the pressure, ν is the kinematic viscosity, and $\mathbf{F}$ represents external forces acting on the fluid, such as gravity or the Coriolis force.

In stratified geophysical flows, where density variations are small compared to the mean density, the Boussinesq approximation is often employed to simplify the Navier-Stokes equation while still capturing the essential dynamics. Under this approximation, the equation becomes

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho_0} \nabla p - \frac{\rho'}{\rho_0} g \hat{\mathbf{z}} + \nu \nabla^2 \mathbf{u} + \mathbf{F}, \tag{3.2}$$

where ρ_0 is the reference density, ρ' is the density perturbation, g is the gravitational acceleration, and $\hat{\mathbf{z}}$ is the unit vector in the vertical direction.

To investigate the dynamics of zonal jets and their interaction with turbulent eddies, we consider the zonal momentum equation under the Boussinesq approximation in a two-dimensional (x, z) domain:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + w\frac{\partial u}{\partial z} = -\frac{1}{\rho_0}\frac{\partial p}{\partial x} - r_m u, \tag{3.3}$$

where u is the zonal velocity, w is the vertical velocity, p is the pressure, ρ_0 is the background density, and r_m is the Rayleigh drag coefficient. We use Rayleigh drag rather than viscosity for consistency with recent studies and as a simple representation of turbulent dissipation

To separate the mean flow from the turbulent eddies, we decompose the flow variables into a zonal mean and a perturbation

$$u = U + u', \quad w = W + w', \tag{3.4}$$

where U and W are the zonal mean velocities, and u' and w' are the perturbation velocities. The zonal mean is defined as

$$U(z, t) = \int u(x, z, t)\, dx, \tag{3.5}$$

and similarly for W.

Substituting the decomposition into the zonal momentum equation, taking the zonal average, and applying periodic boundary conditions in the zonal direction, we arrive at the zonal mean momentum equation

$$\frac{\partial U}{\partial t} = -\frac{\partial}{\partial z}\overline{u'w'} - rU. \tag{3.6}$$

The term $\overline{u'w'}$ is the Reynolds stress, which represents the average zonal momentum flux due to the turbulence. The divergence of the Reynolds stress, $-\frac{\partial}{\partial z}\overline{u'w'}$, acts as a forcing term in the zonal mean momentum equation, driving the evolution of the mean zonal flow U.

The Reynolds stress plays a crucial role in the formation and maintenance of zonal jets, as it represents the interaction between the mean flow and the turbulent eddies. Positive values of the Reynolds stress divergence indicate a convergence of zonal momentum, which accelerates the mean flow, while negative values indicate a divergence of zonal momentum, which decelerates the mean flow.

To close the system of equations and predict the evolution of the mean flow, additional information about the eddy field and its statistical properties is required. One approach is to parameterize the Reynolds stress in terms of the mean flow gradients, using eddy viscosity or diffusivity coefficients. Another approach is to derive additional equations for the higher-order moments of the eddy field, such as the eddy kinetic energy or the eddy enstrophy, and to solve them simultaneously with the zonal mean momentum equation. This approach is known as the moment closure method, and the specific method we use in this study is the Statistical state dynamics, which is based on the moment closure method closed in second order.

Applying Statistical State Dynamics

Statistical state dynamics (SSD) is a powerful framework for studying the behavior of complex, turbulent systems by focusing on the evolution of statistical quantities, such as mean flows and covariance matrices. In the context of zonal jet formation, SSD provides a valuable tool for understanding the interaction between the mean flow and the turbulence and for predicting the emergence and maintenance of zonal jets.

The central idea of SSD is to approximate the eddy-eddy non-linearity in the governing equations by a stochastic forcing term, effectively replacing the deterministic evolution of the eddies with a stochastic process. This approximation allows for the derivation of closed equations for the mean flow and the eddy covariance matrix.

In its full form, the SSD equations for the evolution of the mean flow U and the eddy covariance matrix $\boldsymbol{C}$ can be written as

$$\frac{d\boldsymbol{C}}{dt} = \boldsymbol{A}(U)\boldsymbol{C} + \boldsymbol{C}\boldsymbol{A}(U)^T + \epsilon\boldsymbol{Q}, \tag{3.7}$$

$$\frac{dU}{dt} = R - r_m U, \tag{3.8}$$

$$R = \mathcal{L}(\boldsymbol{C}), \tag{3.9}$$

where R is the Reynolds stress divergence, which is a linear operator $\mathcal{L}$ that maps the covariance matrix $\boldsymbol{C}$ to the Reynolds stress divergence, i.e., $R = \mathcal{L}(\boldsymbol{C})$. The operator $\boldsymbol{A}(U)$ encodes the dynamics of the perturbations and their interaction with the mean flow U, ϵ is a small parameter that controls the strength of the stochastic forcing, and $\boldsymbol{Q}$ is the covariance matrix of the stochastic forcing.

The linear operator $\mathcal{L}$ relates the second-order turbulent statistic, the Reynolds stress divergence R, to the covariance matrix $\boldsymbol{C}$. Specifically, the Reynolds stress divergence is given by

$$R = \frac{1}{4}k(k_+^2 - k_e^2)C_{13}, \tag{3.10}$$

where C_{13} is the covariance between the excited and sheared streamfunction components, $C_{13} = \langle \psi_e \psi_+ \rangle$.

The SSD equations can be further simplified by making the quasi-equilibrium approximation, which assumes that the turbulent eddies quickly adjust to the evolving mean flow. Under this approximation, the time derivative of the covariance matrix is set to zero

$$0 = \boldsymbol{A}(U)\boldsymbol{C} + \boldsymbol{C}\boldsymbol{A}(U)^T + \epsilon\boldsymbol{Q}. \tag{3.11}$$

Solving this equation, which is called the time-independent Lyapunov equation, yields a functional relationship between the mean flow U and the eddy covariance matrix

$$\boldsymbol{C}^* = \boldsymbol{F}(U), \tag{3.12}$$

where $\boldsymbol{F}$ is a matrix-valued function that gives the equilibrium turbulence statistics, $\boldsymbol{C}^*$, as a function of the mean flow structure U.

The resulting SSD equations for the mean flow U can be written as

$$\frac{dU}{dt} = \mathcal{L}[\boldsymbol{F}(U)] - rU, \tag{3.13}$$

which expresses the evolution of the mean flow in terms of the mean flow itself, without explicitly resolving the eddy dynamics.

By focusing on the statistical properties of the flow rather than the detailed dynamics of individual eddies, SSD provides a theoretically tractable framework for investigating the emergence and maintenance of large-scale structures, such as zonal jets, from small-scale turbulence that is autonomous and deterministic and is thus amenable to standard techniques of applied mathematics including stability analysis and bifurcation theory [71].

One of the key advantages of the SSD approach is its ability to analyze zonal jet formation while bypassing the need for long numerical simulations that explicitly characterize turbulent statistics through brute force. This makes SSD a valuable tool for understanding the fundamental mechanisms underlying jet formation and for predicting the properties of jets under different forcing and dissipation conditions.

Linearized SSD Analysis

SSD provides a framework for understanding the evolution of mean flows and turbulence statistics in geophysical flows. A crucial aspect of analyzing such systems is understanding their behavior around equilibrium states, which can be achieved through linearization.

For the standard parameters used in our study, the zonal jet fluctuates around an equilibrium value of $U \approx 0.15$ and $R \approx 0.01$. These values correspond to the fixed point of the SSD system, representing the long-term ($t \to \infty$) state where the system reaches a stable equilibrium.

Recall that the SSD system is described by the following equations:

$$\frac{dU}{dt} = R - r_m U, \tag{3.14}$$

$$\frac{d\boldsymbol{C}}{dt} = \boldsymbol{A}(U)\boldsymbol{C} + \boldsymbol{C}\boldsymbol{A}(U)^T + \epsilon \boldsymbol{Q}, \tag{3.15}$$

$$R = \mathcal{L}(\boldsymbol{C}). \tag{3.16}$$

With he fixed points of this system, denoted by U^*, $\boldsymbol{C}^*$, and R^*, we satisfy the time-independent equations:

$$r_m U^* = R^*, \tag{3.17}$$

$$R^* = \mathcal{L}(\boldsymbol{C}^*), \tag{3.18}$$

$$0 = \boldsymbol{A}(U^*)\boldsymbol{C}^* + \boldsymbol{C}^*\boldsymbol{A}(U^*)^T + \epsilon \boldsymbol{Q}. \tag{3.19}$$

To analyze the system's behavior near these fixed points, we introduce small perturbations $\delta U(t)$ and $\delta \boldsymbol{C}(t)$ around the equilibrium values:

$$U_0 = U^* + \delta U(t), \tag{3.20}$$

$$\boldsymbol{C}_0 = \boldsymbol{C}^* + \delta \boldsymbol{C}(t). \tag{3.21}$$

Linearizing the SSD equations around the fixed point, we obtain a set of linear equations governing the evolution of these perturbations. This linearized system predicts that the perturbations $\delta U(t)$ and $\delta \boldsymbol{C}(t)$ will decay exponentially over time:

$$\delta U = \overline{\delta U} e^{-\lambda^* t}, \tag{3.22}$$

$$\delta \boldsymbol{C} = \overline{\delta \boldsymbol{C}} e^{-\lambda t}, \tag{3.23}$$

$$\overline{R} = \mathcal{L}(\overline{\delta \boldsymbol{C}}) e^{-\lambda t}, \tag{3.24}$$

where $\overline{\delta U}$ is a scalar representing the initial perturbation amplitude of the mean flow, $\overline{\delta \boldsymbol{C}}$ is a matrix representing the initial perturbation of the covariance matrix, and λ

is the decay rate determined by the eigenvalues of the linearized system.

These exponential decays are illustrated in Figure 3.1, which shows the evolution of the mean flow, covariance, and Reynolds stress divergence perturbations after being initialized with random disturbances.

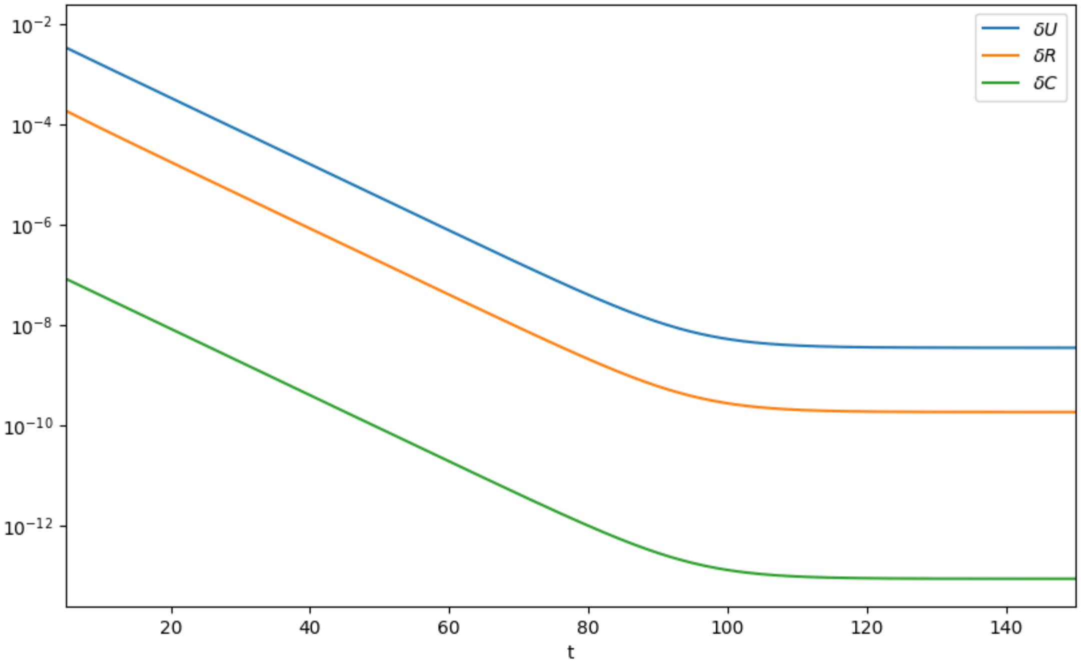

Figure 3.1: Exponential decay of the mean flow perturbation δU, the Reynolds stress divergence perturbation δR, and a representative component of the covariance matrix perturbation $\delta \boldsymbol{C}$, specifically $\delta \boldsymbol{C}_{1,1}$, over time. The y-axis is plotted on a logarithmic scale to highlight the exponential decay.

We are particularly interested in the relationship between the mean flow perturbation $\delta U(t)$ and the Reynolds stress divergence perturbation $\delta R(t)$ during this exponential decay regime.

Using the linearized equations, we can express the ratio between these perturbations as:

$$\frac{U - U^*}{R - R^*} = \frac{\delta U e^{-\lambda t}}{\delta R e^{-\lambda t}} \tag{3.25}$$

$$= \frac{\delta U}{\delta R} \tag{3.26}$$

$$= K = \text{constant}, \tag{3.27}$$

where K is a constant that can be diagnosed from simulations of the linearized system. This constant represents the proportionality between the mean flow and Reynolds stress divergence fluctuations near the equilibrium state.

To determine the value of K, we initialize the linearized system with a small perturbation around the equilibrium state. Specifically, we add a small noise of 5% to the equilibrium values of U^* and $\boldsymbol{C}^*$ obtained from a long-time integration of the SSD system. The system is then allowed to evolve, and we observe the exponential decay of U and R back to their equilibrium values. From the simulations, we find that $K \approx 19.13$.

Once K is known, we can predict the relationship between the Reynolds stress divergence R and the mean flow U in the fluctuating turbulent system using the following equation:

$$R = R^* + \frac{(U - U^*)}{K}. \tag{3.28}$$

This formula provides a linearized approximation of the system's dynamics around the equilibrium state, allowing us to predict the Reynolds stress divergence based on the mean flow fluctuations and the diagnosed constant K.

Challenges in Zonal Jet Studies

The study of zonal jet dynamics faces numerous challenges arising from the complex nature of the underlying physical processes, the limitations of existing theoretical and numerical models, and the scarcity of observational data.

One of the primary difficulties is the multiscale nature of the problem. Zonal jets emerge from the interactions between large-scale mean flows and small-scale turbulent eddies, spanning a wide range of spatial and temporal scales. Capturing the full spectrum of these interactions requires high-resolution numerical simulations that can resolve both the large-scale jet structure and the small-scale eddy dynamics, which can be computationally expensive and may not be feasible for long-term climate studies or for exploring a wide range of parameter spaces.

While statistical state dynamics (SSD) provides a powerful framework, it relies on assumptions and approximations that may not always hold in realistic geophysical flows. The quasi-equilibrium approximation is sometimes used to assume that the turbulent eddies quickly adjust to the evolving mean flow. Another simplification is the representation of eddy-eddy non-linearity, which is often treated as a stochastic forcing term that may not capture the full complexity of the turbulent interactions.

Furthermore, SSD relies on the ergodic assumption, which states that the statistical properties of the turbulent flow, such as the mean and covariance, can be obtained by averaging over either a sufficiently long period or a sufficiently large zonal extent. This assumption may be violated when there are relatively few independent eddies/degrees of freedom across the zonal extent. This occurs, for example, in the Earth's atmosphere, where we only have a handful of eddies in the atmospheric storm track. In these cases, zonal means are still noisy quantities that do not accurately reflect the long-term time average.

The scarcity of observational data also hinders the study of zonal jets. While satellite observations and in-situ measurements have provided valuable insights into the structure and variability of zonal jets in Earth's atmosphere and oceans, the spatial and temporal coverage of these observations is often limited, particularly for the deep ocean and the atmospheres of other planets. This lack of comprehensive observational data makes it challenging to validate and constrain theoretical models and assess the realism of numerical simulations.

To address these challenges, new approaches and tools are essential in the study of zonal jets. Data-driven methods like deep learning have the potential to leverage the vast amounts of data generated by numerical simulations and observations to uncover hidden patterns and relationships that may not be apparent from traditional analysis techniques.

3.2 Deep Learning

Deep learning (DL) has its roots in the broader domain of machine learning (ML), which has been a subject of active research and development for several decades [45, 31]. Machine learning, at its core, involves the development of algorithms and statistical models that enable computer systems to perform specific tasks effectively without being explicitly programmed [11]. However, the limitations of traditional machine learning approaches, particularly in handling complex, high-dimensional data, have paved the way for the emergence of deep learning [68].

The foundational ideas of neural networks can be traced back to the 1940s, with the introduction of the McCulloch-Pitts neuron model [53]. This early work laid the groundwork for the development of the perceptron, a simple linear classifier, by Frank Rosenblatt in 1958 [66].

The perceptron, which mimics the behavior of a single neuron, forms the building block of artificial neural networks. As shown in Figure 3.3, the perceptron takes in weighted inputs, applies an activation function, and produces an output, similar to how a biological neuron processes signals. However, the limitations of single-layer perceptrons, as highlighted by Minsky and Papert in 1969 [54], led to a temporary decline in the popularity of neural networks.

The resurgence of interest in neural networks came with the introduction of backpropagation, a training algorithm that enables the learning of multi-layer networks, by Rumelhart, Hinton, and Williams in 1986 [67]. This breakthrough allowed for the creation of more complex and powerful neural network architectures capable of learning intricate patterns and representations from data.

Deep learning refers to the use of artificial neural networks with multiple layers, as illustrated in Figure 3.2. These deep neural networks can learn hierarchical representations of data, with each layer capturing increasingly abstract features [83]. The ability to learn such representations from raw data, without the need for manual feature engineering, is one of the key advantages of deep learning over traditional machine learning methods.

A crucial property of deep neural networks is their ability to serve as universal function approximators. The Universal Approximation Theorem, proved by Cybenko [19] and Hornik [38], states that a feed-forward neural network with a single hidden

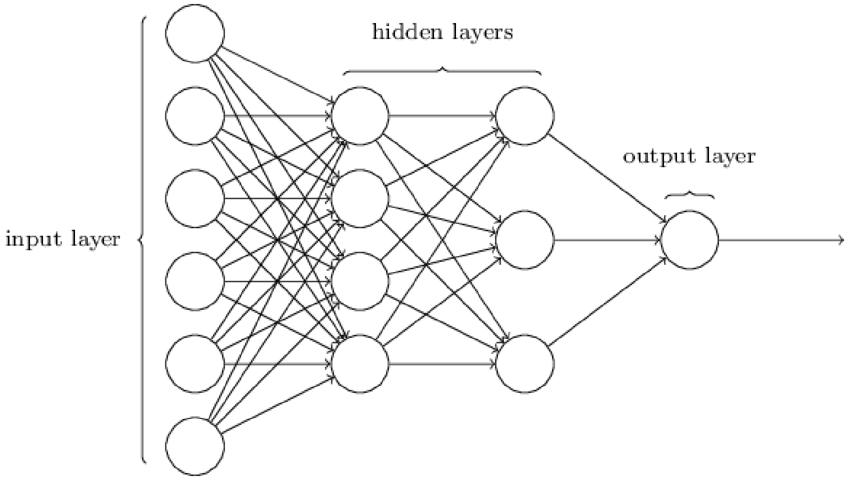

Figure 3.2: A simple multi-layer neural network with an input layer, hidden layers, and an output layer

layer containing a finite number of neurons can approximate any continuous function on compact subsets of $\mathbb{R}^n$, under mild assumptions on the activation function. This theorem provides a theoretical foundation for the expressive power of neural networks and their potential to model complex relationships in data.

The success of deep learning can be attributed to several factors, including the availability of large-scale datasets, advancements in computational hardware (particularly GPUs), and the development of effective training algorithms and regularization techniques [45]. These advancements have enabled the training of deep neural networks with millions of parameters on vast amounts of data, leading to breakthroughs in various domains such as natural language processing [80], speech recognition [36], and reinforcement learning [55]. One of the earliest accomplishments of deep learning is outperforming human experts and traditional computer programs in chess and Go [70]. The ability to learn and master game strategies has captured the public's imagination and showcased the power of deep learning in solving challenging problems.

The ability of deep learning models to automatically learn relevant features from raw data has made them particularly attractive for tackling complex problems in the physical sciences. By leveraging the expressive power of deep neural networks, researchers can uncover hidden patterns, model complex relationships, and make accurate predictions in fields such as quantum mechanics [16], fluid dynamics [15], and climate science [61].

Deep learning offers a promising approach to analyzing and modeling the interactions between large-scale mean flows and small-scale turbulent eddies. By training deep neural networks on extensive datasets obtained from simulations or observational data, we gain insights into the fundamental mechanisms governing the maintenance of zonal jets.

Fundamental Concepts

The landscape of ML is diverse, with a variety of approaches classified based on the nature of the learning signal or feedback available to the system. These include supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning [73]. Deep learning, particularly, excels in scenarios where the complexity and volume of the data surpass the capabilities of traditional machine learning techniques [45].

The term "deep" in deep learning refers to the use of multiple layers in neural networks, which allows these models to learn hierarchical representations of data [68]. This depth enables the network to perform feature extraction at various levels of abstraction, making it particularly powerful for tasks involving unstructured data such as images and text [44].

The core of deep learning is the concept of a neuron, or node, inspired by the biological neuron in the human brain. As shown in Figure 3.3, each neuron receives input, processes it, and passes on its output to the next layer in the network. The input to a neuron is a weighted sum of outputs from neurons in the previous layer, and the weights are the parameters that the network learns during training. The neuron's output is typically by a non-linear activation function to capture complex patterns [31].

Mathematically, the output of a single neuron can be expressed as

$$y = f\left(b + \sum_{i=1}^{n} w_i x_i\right), \tag{3.29}$$

where y is the output of the neuron, f is the activation function, w_i are the weights, x_i are the inputs, b is the bias term, and n is the number of inputs.

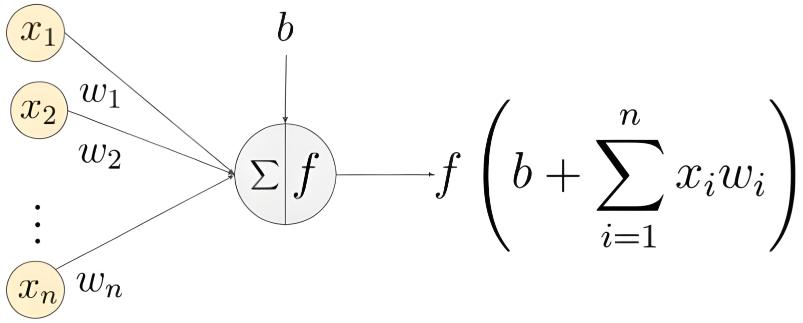

Figure 3.3: A single neuron with inputs $(x_1, x_2, ...x_n)$, its corresponding weights $(w_1, w_2, ...w_n)$, a bias b, and activation function f applied to the weighted sum of the inputs [48]

It's important to choose the right activation function while creating a deep neural network. Some commonly used activation functions include the sigmoid function, hyperbolic tangent (tanh) function, and the rectified linear unit (ReLU) function [56]. The ReLU function, defined as $f(x) = \max(0, x)$, and illustrated in Figure 3.4, has become particularly popular due to its simplicity and effectiveness in deep networks.

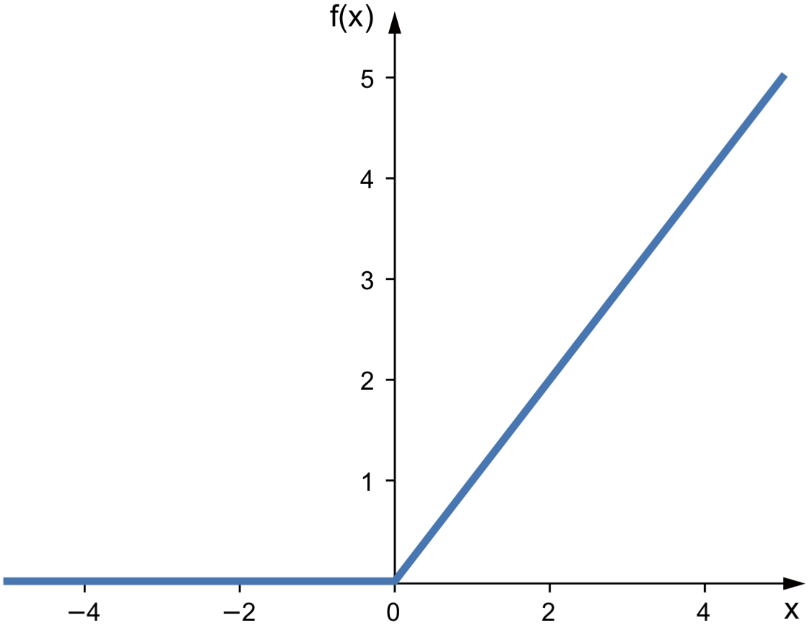

Figure 3.4: The ReLU activation function, $f(x) = \max(0, x)$

Without a non-linear activation function, we would essentially have a linear model, regardless of the number of layers or neurons. The non-linearity allows the network to learn complex, non-linear mappings between the input data and the target output. The ReLU activation function is particularly well-suited for this purpose because it preserves the properties of the linear model for positive inputs while setting negative inputs to zero.

Deep neural networks are composed of multiple layers of interconnected neurons, with each layer learning increasingly abstract representations of the input data. The most basic architecture is the feed-forward neural network (FNN), also known as the multi-layer perceptron (MLP) [66]. In an FNN, the neurons are organized into an input layer, one or more hidden layers, and an output layer. The information flows in a single direction, from the input layer through the hidden layers to the output layer.

The training of a deep neural network involves adjusting the weights and biases to minimize a loss function, which quantifies the discrepancy between the network's predictions and the desired outputs. The choice of loss function is crucial as it directly impacts the model's performance and the learning dynamics.

One of the most commonly used loss functions for regression tasks is the Mean Squared Error (MSE), which is defined as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{3.30}$$

where y_i represents the true value, $\hat{y}_i$ denotes the predicted value by the FNN, and n is the total number of observations.

The optimization process is typically performed using the backpropagation algorithm [67], which efficiently computes the gradients of the loss function with respect to the network's parameters. The gradients are then used to update the parameters using optimization techniques such as stochastic gradient descent (SGD) [12] or its variants, such as the Adam optimizer [42].

The backpropagation algorithm relies on the chain rule of calculus to propagate the gradients from the output layer back to the input layer. For a single neuron, the gradient of the loss function $\mathcal{L}$ with respect to the weight w_i can be expressed as

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial z}\frac{\partial z}{\partial w_i} \tag{3.31}$$

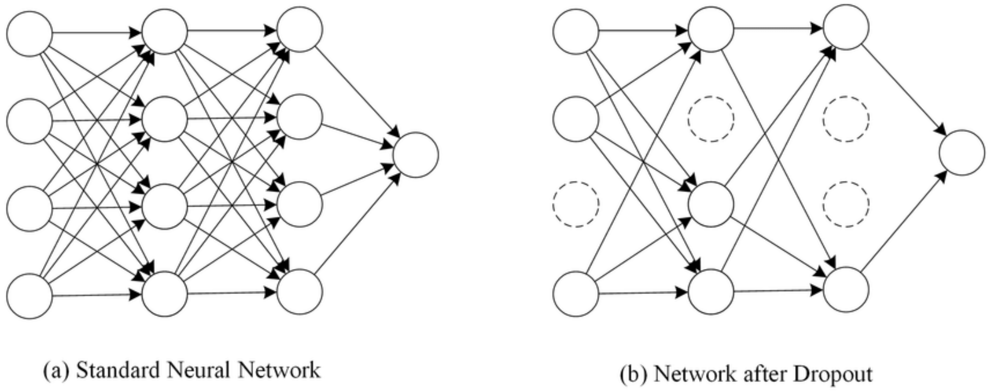

Figure 3.5: (a) is a standard neural network. (b) is the same network after applying Dropout layers. Circles with dotted lines indicate nodes that have been dropped. (Source: [10])

where $z = b + \sum_{i=1}^{n} w_i x_i$ is the weighted sum of the inputs, and $y = f(z)$ is the output of the neuron.

Regularization techniques are employed to prevent overfitting and improve the generalization performance of deep neural networks. Dropout is a particularly effective regularization technique that randomly drops out (sets to zero) a fraction of the neurons during training, which helps prevent the network from relying too heavily on any single neuron or feature. Mathematically, dropout can be expressed as

$$\tilde{y} = m \odot y \tag{3.32}$$

where $\tilde{y}$ is the output of the neuron with dropout applied, m is a binary mask vector sampled from a chosen random distribution, and $\odot$ denotes element-wise multiplication. [31]. Figure 3.5 shows the effects of applying this technique in a standard neural network model.

Batch normalization is a technique that helps accelerate the training of deep neural networks by addressing the internal covariate shift problem. During training, the distribution of inputs to each layer can change as the parameters of the previous layers are updated. This shift in the input distribution, known as the internal covariate shift, can slow down the training process and make it more difficult for the network to converge.

Internal covariate shift occurs because the constant changes in the parameters of earlier layers affect the input distribution of subsequent layers. This can cause gradients to become unstable, leading to slower convergence or even divergence of the training process.

To address this issue, batch normalization normalizes the inputs to each layer by subtracting the batch mean and dividing by the batch standard deviation, followed by a linear transformation with learnable parameters. This ensures that the input distribution remains stable throughout the training process, accelerating convergence and improving the learning process. Additionally, it acts as a form of regularization, reducing the risk of overfitting and enhancing the network's generalization performance.

Motivation for the Deep Learning Approach

Deep learning offers a data-driven approach that complements and extends the capabilities of traditional theoretical methods, such as the statistical state dynamics (SSD). [A diagram illustrating the complementary relationship between deep learning and traditional theoretical methods in studying zonal jets.]

One of the primary advantages of deep learning is its ability to learn from data without the need for explicit programming or a priori assumptions about the underlying physical processes. This is particularly valuable in the study of zonal jets, where the interactions between turbulence, waves, and mean flows can give rise to emergent phenomena that are difficult to identify using traditional approaches such as numerical simulations [79]. By training deep neural networks on large datasets generated by numerical simulations or observational measurements, we can uncover hidden patterns and relationships that may not be apparent from traditional analyses.

Deep learning methods can implicitly account for physics that is neglected in theoretical frameworks such as SSD. This may be crucial for understanding the multiscale nature of zonal jet dynamics, where the interactions between eddies and mean flows, as well as eddy self-interactions, can span a wide range of spatial and temporal scales [79]. By leveraging the hierarchical structure of deep neural networks, we can learn representations of the data that capture these multiscale features and provide a more comprehensive picture of the underlying physical processes.

Another key advantage of deep learning is its ability to handle large and complex datasets. As the resolution of numerical simulations and the volume of observational data continue to increase, traditional theoretical methods may struggle to keep pace with the growing computational demands and complexity of datasets. Deep learning, on the other hand, thrives on large datasets and can efficiently extract meaningful features and patterns from high-dimensional data. This is particularly relevant for studying zonal jets in the context of climate modeling, where the integration of multiple physical processes and the need for long-term simulations can generate massive amounts of data [61].

Moreover, deep learning can help bridge the gap between theoretical models and observational data. By training deep neural networks on a combination of simulated and observed data, we can develop hybrid models that incorporate the strengths of both approaches. These hybrid models can provide a more realistic representation of zonal jet dynamics and help identify discrepancies between theoretical predictions and real-world observations. This is crucial for improving our understanding of the physical mechanisms underlying zonal jet formation and maintenance and for developing more accurate and reliable models of atmospheric and oceanic circulation.

Deep learning should not be seen as a replacement for traditional theoretical methods, but rather as a complementary approach that can provide additional insights and can be used to test existing theories. The SSD, for example, provides a solid foundation for understanding the linear stability of zonal jets and the role of eddy-mean flow interactions in their formation and maintenance [25, 24]. Deep learning can build upon these insights by capturing eddy self-interaction effects and higher-order correlations that are difficult to incorporate into the SSD framework.

Furthermore, deep learning can help guide the development of new theoretical models by identifying key features and relationships in the data that may not be immediately apparent from traditional analyses. By exploring the learned representations of deep neural networks, we can gain new insights into the physical mechanisms underlying zonal jet dynamics and use this knowledge to refine existing theories or develop new ones.

Chapter 4

Methodology

4.1 Overview

This thesis aimed to develop a new approach to investigate the equilibration and maintenance characteristics of zonal jets by implementing deep learning techniques.

The research commenced with an in-depth study of the low-order model (LOM) developed by Fitzgerald and Farrell [27], which provided a simplified yet insightful framework for studying zonal jet behavior in two-dimensional stratified turbulence. The LOM captured several essential interactions between the zonal jet and perturbations, allowing for a more tractable analysis of the underlying mechanisms driving the zonal jet formation and maintenance. This model served as a foundation for generating training datasets and validating the deep learning models developed in this study.

Extensive simulations were performed using the LOM to generate training datasets for the deep learning models. The parameter values and initial conditions were carefully selected based on the findings of previous studies [27, 18] and the physical understanding of zonal jet dynamics to ensure that the generated datasets were representative of the relevant flow regimes.

Statistical analysis was carried out to better understand the data from the LOM simulations. This included the examination of the conditional probability distribution $P(R|U)$, which quantifies the likelihood of observing a particular value of the Reynolds stress (R) given a specific value of the zonal jet velocity (U). The autocorrelation function of the zonal jet velocity U was analyzed to determine the appropriate lag for subsampling the dataset, effectively reducing the temporal resolution of the data while preserving the essential information about the zonal jet dynamics.

A suite of deep learning models tailored for studying zonal jet dynamics was developed and implemented. These models were designed to complement and extend the capabilities of traditional theoretical approaches, such as statistical state dynamics (SSD) [25, 18, 5]. The deep learning models employed in this study included feed-forward neural networks, statistics-informed neural networks (SINN) [85], and physics-informed neural networks (PINNs) [59]. The choice of these architectures was guided by their proven success in modeling complex physical systems and their ability to incorporate prior knowledge and physical constraints into the learning process.

The training of the deep learning models was carried out using appropriate optimization algorithms and loss functions, with careful consideration given to ensuring physical consistency. Extensive validation procedures, such as k-fold cross-validation [31], were employed to assess the models' generalization capabilities and accuracy in predicting zonal jet emergence and evolution. Performance metrics, including the mean squared error (MSE) and the mean absolute error (MAE), were computed to quantify the agreement between the model predictions and the ground truth data. The models' ability to capture the statistical properties of the flow was evaluated to ensure that the learned dynamics were physically consistent with the underlying system.

The results obtained from the trained models were compared with established methods like SSD to demonstrate the complementary nature of the deep learning approach. This comparative analysis highlighted the advantages and limitations of deep learning in capturing the dynamics of zonal jet formation while also identifying areas where the two approaches could complement each other. By combining the insights gained from deep learning with the theoretical foundations provided by SSD and other methods, a more comprehensive understanding of zonal jet dynamics was sought.

The trained deep learning models were subjected to physical interpretation and analysis, to extract insights about the formation and maintenance mechanisms of zonal jets.

4.2 Data Generation and Exploration

Low-Order Model (LOM)

The LOM developed by Fitzgerald and Farrell [27] provides a simplified framework for studying zonal jet dynamics in 2D stratified turbulence. The LOM captures several essential interactions between the zonal jet and perturbations using a severely truncated system that represents the zonal jet and perturbations with a few Fourier components. This simplification allows for a more manageable analysis of the complex dynamics involved and dramatically reduced computational time to simulate the system. The model includes a stochastic excitation term, $\sqrt{\epsilon}\boldsymbol{\xi}$, that drives the system, representing the effects of external forcing and eddy-eddy non-linearity.

To obtain the LOM, Fitzgerald and Farrell [27] chose the stochastic excitation to excite a single standing wave mode so that S is proportional to $\sin(kx)\sin(mz)$. They analyze the interaction between the excited mode and a zonal jet with vertical wavenumber m_U, neglecting the horizontal mean buoyancy, B, as it plays no role in the linear instability responsible for zonal jet formation [27]. The perturbation streamfunction, ψ', the perturbation buoyancy, b', and the zonal jet, U, are written in the form of low-order Fourier truncations as

$$\psi'(x,z,t) = \psi_e \sin(kx)\sin(mz) + \psi_+ \cos(kx)\cos((m+m_U)z), \tag{4.1}$$

$$b'(x,z,t) = b_e \cos(kx)\sin(mz) + b_+ \sin(kx)\cos((m+m_U)z), \tag{4.2}$$

$$U(z,t) = U\sin(m_U z). \tag{4.3}$$

In the original model of Fitzgerald and Farrell [27], the difference wavenumber components were also included, but experimentation showed that including only the sum wavenumbers gave qualitatively similar results. In this work, we retain only the sum of wavenumbers.

The equations of motion for the coefficients are obtained by substituting the expansions (4.1)-(4.3) into the quasi-linear (QL) equations and projecting each term onto the structure functions. For example, the contribution to the ψ_e equation from the mean flow interaction terms in the vorticity equation is given by

$$\left(-\frac{k_e^2}{4}\right)^{-1}\int_0^1 dx\int_0^1 dz(\sin(kx)\sin(mz))(-U\partial_x\Delta\psi'+(\partial_x\psi')U_{zz}) = -\frac{1}{2k_e^2}(k_+^2-m_U^2)U\psi_+, \tag{4.4}$$

where $k_e^2 = k^2 + m^2$ and $k_+^2 = k^2 + (m + m_U)^2$ [27].

The LOM is most compactly written in a vector-matrix form where the state vectors of the excited and sheared wave components are defined as $\boldsymbol{\phi}_e = (\psi_e, b_e)^T$ and $\boldsymbol{\phi}_+ = (\psi_+, b_+)^T$. From this, we obtain

$$\begin{pmatrix}\dot{\boldsymbol{\phi}}_e\\ \dot{\boldsymbol{\phi}}_+\end{pmatrix} = \begin{pmatrix}\boldsymbol{W}_e & 0\\ 0 & \boldsymbol{W}_+\end{pmatrix}\begin{pmatrix}\boldsymbol{\phi}_e\\ \boldsymbol{\phi}_+\end{pmatrix} + U\begin{pmatrix}0 & \boldsymbol{L}_{e,+}\\ \boldsymbol{L}_{+,e} & 0\end{pmatrix}\begin{pmatrix}\boldsymbol{\phi}_e\\ \boldsymbol{\phi}_+\end{pmatrix} + \sqrt{\epsilon}\boldsymbol{\xi}, \tag{4.5}$$

$$\dot{U} = \frac{1}{4}k(k_+^2 - k_e^2)\psi_e\psi_+ - r_m U, \tag{4.6}$$

where ϵ is the energy injection rate and $\boldsymbol{\xi} = (2\sqrt{2}\eta/k_e, 0, 0, 0)^T$, with η being Gaussian white noise with unit variance [27]. The operators W_e and W_+ encode the gravity wave dynamics of the excited and sheared components and are given by

$$W_e = \begin{pmatrix}-1 & k/k_e^2\\ -kN_0^2 & -1\end{pmatrix}, \tag{4.7}$$

$$W_+ = \begin{pmatrix}-1 & -k/k_+^2\\ kN_0^2 & -1\end{pmatrix}. \tag{4.8}$$

The operators $L_{e,+}$ and $L_{+,e}$ encode the interactions between the zonal jet and the perturbations and are given by

$$L_{e,+} = \begin{pmatrix}-\frac{k}{2k_e^2}(k_+^2 - m_U^2) & 0\\ 0 & \frac{k}{2}\end{pmatrix}, \tag{4.9}$$

$$L_{+,e} = \begin{pmatrix}-\frac{k}{2k_+^2}(m_U^2 - k_e^2) & 0\\ 0 & -\frac{k}{2}\end{pmatrix}. \tag{4.10}$$

Equation (4.5) is the LOM analog of the eddy evolution equation. The LOM demonstrates that zonal jets can form spontaneously under certain parameter conditions due to feedback between the zonal jet and the perturbation statistics.

For a simpler representation, the LOM can be written as a set of coupled ordinary differential equations (ODEs):

$$\frac{d\boldsymbol{s}}{dt} = A(U)\boldsymbol{s} + \sqrt{\epsilon}\boldsymbol{S}, \tag{4.11}$$

$$\frac{dU}{dt} = R - rU, \tag{4.12}$$

$$R = \text{constant} \times \psi_e \psi_+. \tag{4.13}$$

Here, $\boldsymbol{s} = (\psi_e, b_e, \psi_+, b_+)$ represents the state vector of the excited and sheared components of the streamfunction and buoyancy, as defined earlier. The linear operator $A(U)$ encodes the dynamics of the perturbations and their interaction with the zonal jet, U.

The LOM also demonstrates that zonal jets can form in stochastically excited flows in which the vertical wavenumber of the zonal jet is not contained in the excitation spectrum [27]. This finding highlights the ability of the LOM to capture the essential interactions between the zonal jet and perturbations, even when the excitation spectrum does not directly force the zonal jet wavenumber.

The simplicity and ability to reproduce key features make it an attractive framework for theoretical and computational investigations. However, as a severely truncated system, the LOM does not capture all details present in more comprehensive models or real-world flows. Additionally, the LOM does not accurately model the finite-amplitude equilibration of the zonal jet, which is an important aspect of zonal jet dynamics in nature.

Deep learning techniques can be leveraged to build upon the LOM and gain new insights into the physics of zonal jet formation and maintenance. By integrating deep learning with the LOM, we can develop a proof of concept for understanding the full systems and discover ways to effectively incorporate deep learning into these systems.

One potential avenue for integrating deep learning with the LOM is to use neural networks to learn the underlying dynamics of the system from data generated by the model. By training a neural network on the time series of the zonal jet, U, and the perturbation fields, ψ' and b', we can develop a data-driven model that captures the essential features of the LOM's dynamics.

Deep learning can also be used to explore the parameter space of the LOM more efficiently. By training a neural network to predict the behavior of the system under different parameter combinations, we can identify the regions of parameter space that are most likely to exhibit interesting or important dynamics, such as the spontaneous formation of zonal jets. This can help guide future theoretical and computational investigations, allowing researchers to focus their efforts on the most promising areas of study.

Ultimately, the goal of integrating deep learning with the LOM is to develop a framework for understanding the full systems of zonal jet dynamics in a more transparent and interpretable way. By using the LOM as a proof of concept, we can demonstrate the potential of deep learning to uncover the underlying physics of these complex systems and pave the way for more effective integration of deep learning into comprehensive models of geophysical and astrophysical fluid dynamics.

Dataset Preparation

Solving the LOM using the Euler-Maruyama Method

To generate the dataset for studying zonal jet dynamics, we use the Euler-Maruyama method, a numerical scheme designed for solving stochastic differential equations (SDEs). SDEs are differential equations that incorporate a stochastic term, representing the influence of random or unpredictable forces on the system's dynamics.

The Euler-Maruyama method is an extension of the classical Euler method, which is used for solving ordinary differential equations (ODEs). In the case of SDEs, an additional term is introduced to account for the stochastic component, typically modeled as a Wiener process or Brownian motion.

Mathematically, the Euler-Maruyama method approximates the solution of an SDE of the form:

$$dX_t = a(X_t, t)\, dt + b(X_t, t)\, dW_t \tag{4.14}$$

where X_t is the state variable, $a(X_t, t)$ is the deterministic drift term, $b(X_t, t)$ is the diffusion term, and dW_t represents the increments of the Wiener process.

The Euler-Maruyama method discretizes the time interval $[0, T]$ into N equal subintervals of width $\Delta t = T/N$. Starting from an initial condition $X_0 = x_0$, the state variable X_t is updated at each time step using the following recursive formula:

$$X_{n+1} = X_n + a(X_n, t_n)\, \Delta t + b(X_n, t_n)\, \Delta W_n \tag{4.15}$$

where $\Delta W_n = W_{t_{n+1}} - W_{t_n}$ are independent and identically distributed normal random variables with mean zero and scaled with $\sqrt{\Delta t}$.

The presence of the term $\sqrt{\Delta t}$ in the ΔW_n is crucial for ensuring the correct scaling of the random increments. This scaling arises from the properties of the Wiener process, which has independent and stationary increments with a variance proportional to the time interval.

Unlike deterministic numerical methods, the Euler-Maruyama method introduces randomness into the solution, capturing the stochastic nature of the underlying system. This randomness is essential for accurately modeling phenomena influenced by random or unpredictable forces, such as turbulence or external noise.

The accuracy of the Euler-Maruyama method depends on the choice of the time step Δt. Smaller time steps generally lead to more accurate solutions but also increase computational costs. Higher-order methods, such as the Milstein method, can provide better accuracy but at the expense of increased complexity.

Calculating the Conditional Probability

The conditional probability $P(R|U)$ has been a key part of our datasets and the statistical analysis of the relationship between the Reynolds stress R and the zonal jet velocity U in the LOM. It quantifies the likelihood of observing a particular value of the Reynolds stress given a specific value of the zonal jet velocity, providing insights into the statistical dependence between these two variables.

Mathematically, the conditional probability $P(R|U)$ is defined as the ratio of the joint probability distribution $P(R, U)$ to the marginal probability distribution $P(U)$ [57, 29]:

$$P(R|U) = \frac{P(R,U)}{P(U)}, \tag{4.16}$$

where $P(R,U)$ represents the probability of observing a specific combination of R and U values, and $P(U)$ represents the probability of observing a specific value of U, regardless of the value of R.

In practice, the conditional probability $P(R|U)$ is estimated from the LOM dataset by constructing a 2D histogram of the Reynolds stress and zonal jet velocity values. The range of U values is divided into a set of N_U equally spaced bins, denoted as $\{U_1, U_2, \ldots, U_{N_U}\}$, and the range of R values is divided into a set of N_R equally spaced bins, denoted as $\{R_1, R_2, \ldots, R_{N_R}\}$.

The 2D histogram is then constructed by counting the number of data points falling into each bin defined by the U and R intervals. Let n_{ij} denote the count of data points in the bin corresponding to the i-th U bin and the j-th R bin. The joint probability $P(R_j, U_i)$ is estimated as

$$P(R_j, U_i) = \frac{n_{ij}}{N}, \tag{4.17}$$

where N is the total number of data points in the LOM dataset.

The marginal probability $P(U_i)$ is estimated by summing the counts across all R bins for each U bin

$$P(U_i) = \sum_{j=1}^{N_R} P(R_j, U_i) = \frac{\sum_{j=1}^{N_R} n_{ij}}{N}. \tag{4.18}$$

Finally, the conditional probability $P(R_j|U_i)$ is computed by dividing the joint probability $P(R_j, U_i)$ by the marginal probability $P(U_i)$

$$P(R_j|U_i) = \frac{P(R_j, U_i)}{P(U_i)} = \frac{n_{ij}}{\sum_{j=1}^{N_R} n_{ij}}. \tag{4.19}$$

The resulting conditional probability distribution $P(R|U)$ characterizes the statistical relationship between the Reynolds stress and the zonal jet velocity. By examining how the R distribution varies across different U bins, we can gain insights into the nature of the turbulent momentum flux and its role in the formation and maintenance of the zonal jets [27, 25, 18, 5].

The conditional probability $P(R|U)$ is a valuable target variable for training models to learn the statistical relationship between the Reynolds stress and the zonal jet velocity. By incorporating it into the loss function or the model architecture, we can guide the learning process towards capturing the essential features of the turbulent momentum flux and its dependence on the large-scale flow [31, 11].

For example, a deep learning model can be trained to predict the conditional probability distribution $P(R|U)$ given the zonal jet velocity U as input. The model can be implemented to produce a probability distribution over the R bins for each input U value. By minimizing the loss function during training, the deep learning model can learn to accurately reproduce the statistical relationship between R and U as captured by the conditional probability distribution $P(R|U)$. The trained model can then be used to make predictions of the Reynolds stress distribution for new values of the zonal jet velocity, providing a powerful tool for investigating the feedback mechanisms between turbulence and large-scale flows in the LOM.

Autocorrelation & Subsampling

The autocorrelation function is used to analyze the temporal coherence and memory of the time series of the zonal jet velocity U in the LOM. It quantifies the similarity between the signal and a delayed version of itself as a function of the lag, providing insights into the persistence and periodicity of the underlying dynamics [13].

The autocorrelation of U is calculated to determine the appropriate lag for subsampling the dataset. The goal is to identify the lag at which the autocorrelation function first crosses a specified threshold, such as $1/e$, indicating a significant decrease in the correlation between data points separated by this lag.

To compute the autocorrelation function, we first calculate the fluctuations of the zonal jet velocity around its mean value

$$U'(t) = U(t) - \bar{U}, \tag{4.20}$$

where $U'(t)$ represents the fluctuations, $U(t)$ is the zonal jet velocity at time t, and $\bar{U}$ is the mean value of U over the entire time series.

The autocorrelation function $R(\tau)$ is then computed as

$$R(\tau) = \frac{1}{N-\tau} \sum_{t=1}^{N-\tau} U'(t)U'(t+\tau), \tag{4.21}$$

where τ is the lag, and N is the total number of data points in the time series.

To ensure that the autocorrelation function ranges between -1 and 1, we normalize it by the variance of the fluctuations

$$\rho(\tau) = \frac{R(\tau)}{R(0)}, \tag{4.22}$$

where $\rho(\tau)$ is the normalized autocorrelation function, and $R(0)$ is the autocorrelation at zero lag, which is equal to the variance of the fluctuations.

The optimal lag for subsampling the dataset is determined by finding the lag at which the normalized autocorrelation function first crosses a specified threshold, such as $1/e$. This lag, denoted as τ_c, is then found by observing the intersection point.

With the critical lag τ_c determined, the dataset is subsampled by selecting every τ_c-th data point, effectively reducing the temporal resolution of the data while preserving the essential information about the zonal jet dynamics [41]. By doing this, the computational burden associated with training deep learning models on large datasets was also reduced as it removes redundant data points that do not contribute significantly to the learning process.

Moreover, subsampling based on the autocorrelation function ensures that the dataset captures the relevant time scales of the zonal jet dynamics since it is determined by the intrinsic properties of the system rather than an arbitrary choice of subsampling interval [41].

4.3 Baseline Models and Proof of Concept

Before the development and implementation of our deep learning models, it was essential to establish a proof of concept that demonstrates the feasibility of learning patterns and performing prediction tasks on the LOM data. Traditional ML models, such as XGBoost and linear regression, were used as baseline models [17, 35].

Traditional Machine Learning Models

Gradient Boosted Random Forest

XGBoost, short for eXtreme Gradient Boosting, is a powerful machine learning technique that combines multiple simple models (called weak learners) to create a strong predictive model. In this case, the weak learners are decision trees, which are a type of model that makes predictions by following a series of rules based on the input features [17].

The core idea behind XGBoost is to train a sequence of decision trees, where each subsequent tree is trained to correct the mistakes made by the previous trees. This is achieved by minimizing an objective function that measures the difference between the predicted values and the actual values while also penalizing overly complex models to prevent overfitting [17].

Mathematically, the objective function of XGBoost can be expressed as

$$\mathcal{L}(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k). \tag{4.23}$$

Here, $\mathcal{L}(\phi)$ is the objective function to be minimized, $l(y_i, \hat{y}_i)$ measures the difference between the true value y_i and the predicted value $\hat{y}_i$ for the i-th data point, $\Omega(f_k)$ penalizes the complexity of the k-th decision tree f_k, and K is the total number of trees in the ensemble [17].

XGBoost uses a technique called gradient boosting to minimize the objective function. At each iteration, the algorithm computes the gradients of the loss function with respect to the predicted values and uses these gradients to guide the construction of the next decision tree. The new tree is then added to the ensemble, and the process is repeated until a specified number of trees is reached or a certain performance criterion is met [17].

Linear Regression

Linear regression is a simple yet powerful machine learning algorithm that aims to find the best-fitting linear relationship between the input features and the target variable. In other words, it tries to find a straight line (or a plane in higher dimensions) that best describes the relationship between the input features and the target variable [35, 51].

The linear regression model can be expressed as

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon, \tag{4.24}$$

where y is the target variable, $x_1, x_2, \ldots, x_p$ are the input features, $\beta_0, \beta_1, \ldots, \beta_p$ are the coefficients that define the linear relationship, and ϵ is an error term that accounts for the noise or unexplained variation in the data.

The coefficients $\beta_0, \beta_1, \ldots, \beta_p$ are estimated using a method called least squares, which minimizes the sum of squared differences between the predicted values and the actual values in the training data.

4.4 Deep Learning Models

In our exploration, we developed and implemented many deep learning models on the datasets initially created. In this section, we will talk about some of the most promising models we worked with, such as Feed-forward Neural Networks, Statistics-Informed Neural Networks, and Physics-Informed Neural Networks.

4.4.1 Feed-forward Neural Networks

Feed-forward Neural Network (FNN) is the simplest kind of artificial neural network architecture, and it serves as the foundation for understanding more complex deep learning models. In this model, information moves in only one direction—forward—from the input nodes, through the hidden nodes (if any), and finally to the output nodes. There are no cycles or loops in this network, hence the term "feed-forward."

The architecture of an FNN consists of three main types of layers: the input layer, one or more hidden layers, and the output layer. The input layer receives the initial data, the hidden layers process the data by performing weighted sums and applying activation functions, and the output layer produces the final predictions.

In our work with this model, the dataset generated by the LOM is used as the training data where the input features are derived from the zonal jet flow, and the output is the Reynolds stress divergence, indicative of turbulence. The primary objective of the FNN is to discern the intricate mapping from these inputs to the corresponding outputs, thereby encapsulating the complex interplay between the zonal jet flow and the emergent turbulence.

To introduce non-linearity into the model—essential for capturing the non-linear dynamics of the system—we use the ReLU activation function. The model's performance is then quantified using the MSE as the loss function.

We also incorporate normalization layers that standardize the input features to have a mean of zero and a standard deviation of one. This normalization is critical as it ensures that no single feature disproportionately influences the model's learning due to its scale. Dropout layers are integrated into the FNN to mitigate the risk of overfitting. During the training phase, these dropout layers randomly deactivate a subset of neurons, compelling the network to develop more robust features that are not overly reliant on any individual neuron [31].

The batch size is shaped by the lag value derived from the autocorrelation analysis, which enhances both the stability of the gradient descent and the convergence rate during the training process.

For optimization, we utilize the Adam optimizer, facilitating a more efficient convergence to the optimal set of weights within the network [31].

Our FNN architecture is designed with multiple hidden layers to support the network's ability to model the sophisticated relationship between the zonal jet flow and turbulence. The depth of the network and other hyperparameters, such as the number of neurons in each layer and the learning rate, all undergo a process of trial-and-error and thorough fine-tuning to refine the model's performance and predictive capabilities.

Adding Physical Constraints

While the MSE is a common choice for a loss function, it cannot directly capture the underlying physical relationships between the input and output variables. To address this limitation and incorporate physical grounding into the data-driven model, we introduce a custom loss function that combines the MSE loss with an additional term related to the conditional probability $P(R|U)$.

The custom loss function is defined as:

$$\mathcal{L} = \mathrm{MSE}(R, \hat{R}) + \lambda \int_R |1 - P(R|U)| dR, \tag{4.25}$$

where $\mathrm{MSE}(R, \hat{R})$ is the standard mean squared error between the true Reynolds stress divergence R and the predicted value $\hat{R}$, $P(R|U)$ is the conditional probability of observing R given the zonal jet mean flow U, and λ is a weighting factor that controls the relative importance of the integral penalty term.

The integral penalty term, $\int_R |1 - P(R|U)| dR$, measures the absolute difference between the predicted conditional probability $P(R|U)$ and the desired behavior, which is typically informed by the underlying physics of the problem. By minimizing this term, the model is encouraged to learn representations that align with the expected physical relationships between the input and output variables.

The integral penalty term acts as a regularizer, penalizing deviations from the desired conditional probability distribution and guiding the model towards solutions that are consistent with the known physical principles governing the zonal jet dynamics.

From a mathematical perspective, the custom loss function can be interpreted as a combination of two objectives: minimizing the mean squared error to ensure accurate predictions of the Reynolds stress divergence and minimizing the integral penalty term to enforce consistency with the expected conditional probability distribution.

4.4.2 Statistics-Informed Neural Networks

In situations where the underlying governing equations are unknown or difficult to specify, Statistics-Informed Neural Networks (SINNs) provide a data-driven approach to modeling stochastic dynamical systems, just like our system. The key innovation of the SINN framework lies in its loss function design. Rather than seeking a direct pathwise approximation of the stochastic dynamics, the SINN constructs simulated trajectories that align with the example trajectories in terms of their statistical properties. This is achieved by incorporating terms in the loss function that penalize the differences between the predicted and true statistical characteristics, such as the probability density function (PDF) and autocorrelation function (ACF) of the target variable.

The theoretical foundation of SINN is based on the universal approximation theorem for stochastic systems and the projection-operator formalism for stochastic modeling [85]. The universal approximation theorem states that, under certain conditions, a recurrent neural network (RNN) with Gaussian white noise as input can universally approximate arbitrary stochastic systems.

This theorem provides a theoretical justification for the modeling capacity of stochastic RNNs and the approach of embedding random noise into the system. By leveraging this theoretical insight, SINNs employ Long Short-Term Memory (LSTM) networks, a specific type of RNN, to capture non-Markovian memory effects that the underlying stochastic system might exhibit.

The SINN architecture consists of a multi-layer LSTM component to learn the temporal dynamics of stochastic processes and a dense layer attached to the output gate of the LSTM as a “read-out” device. The model uses a stream of independent and identically distributed (i.i.d.) random numbers as input to the model. This input serves as the entropy source, allowing the SINN to generate different realizations of the stochastic process through deterministic operations.

The loss function used to train SINNs incorporates terms that penalize the difference between the predicted and true statistical properties of the target variable, such as the PDF and ACF. The loss function includes terms that penalize the difference between the predicted and true PDF and ACF

$$L(\theta) = \lambda_1 ||PDF(h(W_t;\theta)) - PDF(X_t)||^2 + \lambda_2 ||ACF(h(W_t;\theta)) - ACF(X_t)||^2, \quad (4.26)$$

where θ represents the trainable parameters of the SINN, $h(W_t;\theta)$ represents the SINN’s prediction of the target variable X_t, and λ_1 and λ_2 represent weighting factors that control the relative importance of the PDF and ACF terms.

By minimizing this loss function, the SINN learns to accurately reproduce the statistical behavior of the target stochastic process, capturing the complex interactions and temporal dependencies inherent in the system.

4.4.3 Physics-Informed Neural Network

Now, while the simple FNN model and the SINN model rely solely on data to learn the underlying patterns and relationships, the Physics-Informed Neural Network (PINN), introduced by Raissi et al. in 2019 [59], incorporates additional physical constraints into the models in the form of differential equations. This method has gained significant attention due to its ability to incorporate physical laws governed by partial differential equations (PDEs) or ordinary differential equations (ODEs) into the learning process of neural networks.

By incorporating the governing equations of a system into the loss function of the neural network, PINNs ensure that the learned solution satisfies the physical laws of the system, leading to more accurate and physically consistent predictions. This approach also allows for the solution of inverse problems, where some parameters or terms in the governing equations are unknown and need to be inferred from the data.

The key idea behind PINNs is to represent the solution of the governing equations using a neural network and train the network to minimize a loss function that consists of two parts: the data loss and the physics-informed loss. The data loss measures the discrepancy between the network predictions and the available data, while the physics-informed loss quantifies the residual of the governing equations.

Let's consider a general form of a system of ODEs:

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(t, \mathbf{u}; \lambda), \tag{4.27}$$

where $\mathbf{u}(t) = [u_1(t), u_2(t), ..., u_n(t)]$ is the vector of state variables, t is the independent variable (e.g., time), $\mathbf{f}$ is a vector-valued function representing the right-hand side of the ODEs, and λ denotes the parameters of the system.

To solve our LOM using PINNs, we approximate the state variables and the zonal jet using a neural network $\mathbf{u}_\theta(t)$, where θ represents the network parameters (weights and biases). The network takes the independent variable t as input and outputs the approximated state variables and zonal jet.

The loss function for training the PINN consists of two parts:

1. Data loss:

$$\mathcal{L}_{\text{data}} = \frac{1}{N_d} \sum_{i=1}^{N_d} \left\| \mathbf{u}_\theta(t_i) - \mathbf{u}_i \right\|^2 , \tag{4.28}$$

where N_d is the number of available data points, $\mathbf{u}_i$ is the observed value of the state variables and zonal jet at time t_i, and $\|\cdot\|$ denotes the Euclidean norm.

2. Physics-informed loss:

$$\mathcal{L}_{\text{physics}} = \frac{1}{N_f} \sum_{i=1}^{N_f} \left\| \frac{d\mathbf{u}_\theta}{dt}(t_i) - \mathbf{f}(t_i, \mathbf{u}_\theta(t_i); \lambda) \right\|^2 , \tag{4.29}$$

where N_f is the number of collocation points, and $\frac{d\mathbf{u}_\theta}{dt}(t_i)$ is computed using automatic differentiation.

The total loss is then defined as a weighted sum of the data loss and the physics-informed loss:

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \alpha \mathcal{L}_{\text{physics}}, \tag{4.30}$$

where α is a hyperparameter that balances the contribution of the two loss terms.

The network parameters θ are learned by minimizing the total loss using gradient-based optimization algorithms like the Adam optimizer [59].

4.4.4 Model Training and Validation

Effective training and validation of deep learning models are essential for ensuring their accuracy, generalization ability, and reliability in capturing the underlying dynamics of complex systems. Now, we will focus on the key components of the training process to discuss their impacts.

Optimization Algorithms

Optimization algorithms are responsible for updating the model's parameters (weights and biases) iteratively to minimize the defined loss function. One of the most widely used optimization algorithms in deep learning is Stochastic Gradient Descent (SGD) [65]. SGD updates the model parameters in the direction of the negative gradient of the loss function, computed using a randomly selected subset of the training data (mini-batch).

However, SGD can be sensitive to the choice of the learning rate and may converge slowly, especially for complex problems. This is where the Adam optimizer plays its part [42].

Adam combines the advantages of two other popular optimization algorithms, AdaGrad [21] and RMSProp [76], by adapting the learning rate for each parameter based on the first and second moments of the gradients.

It has been shown to perform well in the training of PINNs [59] since its adaptive learning rate and momentum-like behavior help in quickly converging the models on hard problems.

Loss Functions

In a simple FNN, the loss function measures the discrepancy between the model's predictions and the available data. Two commonly used data loss functions are the Mean Squared Error (MSE) and the Mean Absolute Error (MAE).

The MSE loss is defined as:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{4.31}$$

where N is the number of data points, y_i is the true value, and $\hat{y}_i$ is the model's prediction.

The MAE loss is defined as:

$$\mathcal{L}_{\text{MAE}} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i|. \tag{4.32}$$

The MSE loss penalizes larger errors more heavily due to the squared term, while the MAE loss treats all errors equally. The choice between MSE and MAE depends on the specific problem and the desired properties of the solution. For our case, the MSE loss is more suitable since we want to emphasize the importance of accurately capturing the larger-scale features of the zonal jet dynamics. The MAE loss is sometimes preferred when we aim to obtain a solution that is less sensitive to outliers and noise in the data.

Activation Functions

Activation functions introduce non-linearity into the model, enabling it to learn complex patterns and relationships in the data. Some commonly used activation functions in deep learning include:

1. Sigmoid: $\sigma(x) = \frac{1}{1+e^{-x}}$

2. Hyperbolic Tangent (Tanh): $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

3. Rectified Linear Unit (ReLU): $\text{ReLU}(x) = \max(0, x)$

4. Leaky ReLU: $\text{LeakyReLU}(x) = \max(\alpha x, x)$, where α is a small positive constant.

The sigmoid and tanh functions are bounded and differentiable, but they can suffer from the vanishing gradient problem. The ReLU function, on the other hand, is unbounded and has a constant gradient for positive inputs, which helps in alleviating the vanishing gradient problem. However, ReLU can lead to ”dead” neurons if the inputs are consistently negative. The Leaky ReLU function addresses this issue by allowing a small negative slope for negative inputs.

For our FNN models, using ReLU in the hidden layers helps in learning the nonlinear dynamics of the zonal jet. However, for the PINN model, since the physics-informed loss involves computing the derivatives of the network's output with respect to the input (time), using activation functions that are continuously differentiable, such as tanh or sigmoid, leads to more stable and accurate gradient calculations.

Hyperparameters

Hyperparameters are the settings that control the learning process of a deep learning model, such as the learning rate, batch size, and number of epochs:

1. Learning Rate: The learning rate determines the step size at which the model's parameters are updated in the direction of the negative gradient of the loss function. A higher learning rate can lead to faster convergence but may also cause the model to overshoot the optimal solution and oscillate. A lower learning rate can result in slower convergence but may help in finding a more stable and accurate solution. In practice, the learning rate is often tuned using techniques such as learning rate scheduling, where the learning rate is gradually decreased over the course of training.

2. Batch Size: The batch size refers to the number of training examples used in each iteration of the optimization algorithm. Larger batch sizes can lead to more stable gradient estimates and faster convergence, but they also require more memory and computational resources. A smaller batch size can introduce more noise in the gradient estimates but may help escape local minima and regularize the model. The choice of the batch size often depends on the available computational resources and the specific problem at hand.

3. Number of Epochs: An epoch is a complete pass through the entire training dataset. The number of epochs determines how many times the model is exposed to the training data during the learning process. A higher number of epochs can lead to better fitting of the data but may also cause overfitting if the model starts to memorize the training examples. A lower number of epochs may result in underfitting, where the model fails to capture the underlying patterns in the data. The optimal number of epochs can be determined using techniques such as early stopping, where the training is stopped when the model's performance on a validation set starts to degrade.

Other important hyperparameters include the number of hidden layers and neurons in the neural network, the regularization techniques (e.g., L1 or L2 regularization), and the weighting factors of the loss functions. These hyperparameters can be tuned using techniques such as trial-and-error or grid search, where different combinations of hyperparameters are evaluated and the best-performing combination is selected.

Model Validation: Assessing Model Performance

Evaluating the performance of our deep learning models is crucial to ensure their effectiveness and reliability. Before discussing any of the results, let's understand simply the process of assessing the quality of neural network predictions, both visually and through quantitative metrics.

Visual Assessment

We can visually compare the predicted outputs with their true values to evaluate a model's performance by plotting the true input values against the predicted neural network outputs, as well as the true input values against the true output values.

If the "True Input vs. Predicted Output" plot closely resembles the "True Input vs. True Output" plot, it indicates that the neural network has effectively learned the underlying patterns and relationships within the system. As the difference between these two plots diminishes, it suggests that the neural network is acting as an accurate solver for the system, capturing the dynamics with high fidelity.

Quantitative Assessment

Quantitative metrics offer a more objective and precise evaluation of a neural network's performance. Two commonly used metrics in this context are the Mean Squared Error (MSE) and the R-squared (R^2) value.

A lower MSE value indicates that the neural network's predictions are closer to the true values, suggesting better performance. Conversely, a higher MSE value implies larger deviations between the predicted and true values, indicating poorer performance.

The R^2 value measures the proportion of the variance in the true values that can be explained by the neural network's predictions. It ranges from 0 to 1, with a value closer to 1 indicating a better fit between the predicted and true values.

A higher R^2 value indicates that the neural network's predictions are better able to capture the variability in the true values, suggesting a more accurate representation of the underlying patterns of the system.

Monitoring Training Progress

It is also essential to monitor the training progress of a neural network to ensure that it is learning effectively and avoiding issues such as overfitting or underfitting. This can be achieved by plotting the training loss and validation loss over the course of training epochs.

The training loss represents the error or difference between the neural network's predictions and the true target values during the training process. The validation loss, on the other hand, represents the error or difference between the predictions and the true values on a separate validation dataset.

Ideally, both the training loss and validation loss should decrease and converge to a low value, indicating that the neural network is learning the patterns in the data effectively and making accurate predictions. However, if the training loss continues to decrease while the validation loss starts to increase, it suggests that the model is overfitting to the training data and may not generalize well to unseen data.

Probability Distributions: Color Plots

Since the relationship between input variables (e.g., jet strength) and output variables (e.g., turbulence) can be complex and nonlinear, visualizing the probability distributions of the true and predicted outputs can provide valuable insights into the neural network's performance.

These color plots represent the relationship between the input variable (e.g., jet strength) and the output variable (e.g., turbulence), with the color scale indicating the probability density or "Probability Density" of the data points. By comparing the color plot of the true output with the color plot of the predicted output, we can assess how well the neural network has learned to capture the underlying patterns and distributions.

If the predicted output plot closely resembles the true output plot, with the scatter points aligning well and the color patterns being similar, it indicates that the neural network has effectively learned the intricate relationships and probability distributions within the system.

Prediction Quality: Predicted vs. True Values

Another valuable tool for evaluating the performance of a neural network is the scatter plot of predicted values versus true values accompanied by a 45-degree reference line.

In these plots, each point represents an individual data point, with the x-axis representing the true values and the y-axis representing the corresponding predicted values by the neural network. The 45-degree line represents the ideal scenario where the predicted values perfectly match the true values.

If the scatter points cluster closely around the 45-degree line, it indicates that the neural network is making accurate predictions and effectively capturing the underlying patterns in the data. Conversely, if the points deviate significantly from the 45-degree line, it suggests that the neural network is struggling to learn the true relationships and patterns within the system.

Chapter 5

Results and Discussion

This chapter presents the results of applying deep learning techniques to study zonal jet dynamics with the low-order model (LOM) of two-dimensional stratified turbulence. We first establish a baseline understanding of the system's dynamics using linear regression and XGBoost models. These models demonstrate the feasibility of learning patterns and performing prediction tasks on the LOM data.

We then discuss the findings of the deep learning models, including feed-forward neural networks (FNNs), statistics-informed neural networks (SINNs), and physics-informed neural networks (PINNs). These models were designed to complement and extend the capabilities of traditional theoretical approaches like statistical state dynamics (SSD) by leveraging the vast amounts of data generated by LOM simulations. We discuss the findings and performance evaluation of each model, highlighting their strengths and limitations in capturing the dynamics of zonal jets.

Finally, we compare the performance of the different models, emphasizing the complementary nature of the deep learning approach to traditional methods. We discuss the implications of our findings for understanding zonal jet dynamics and the potential for future research in this area.

5.1 Baseline Models: Proof of Concept

5.1.1 Linear Regression Model

The linear regression model is used as a baseline model to understand the relationship between the input variable (U) and the target variable (zonal jet dynamics). This approach provides a starting point for further analysis and model refinement.

Model Performance

The model's performance was evaluated using the Mean Squared Error (MSE) and the Coefficient of Determination (R-squared, R^2). The MSE measures the average squared difference between predicted and true values, while the R^2 value represents the proportion of variance in the target variable explained by the model.

Metric	Value
MSE	0.0054
R^2	0.0014

Table 5.1: Performance metrics for the linear regression model.

The linear regression model achieved an MSE of 0.0054 and an R^2 value of 0.0014. The very low R^2 value indicates that the linear model explains only a negligible portion of the variance in turbulence (R), suggesting a poor fit to the data.

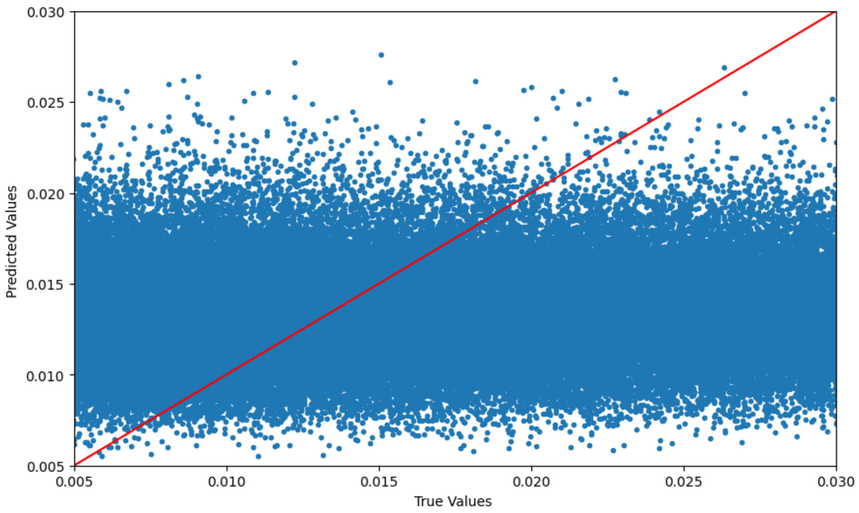

Figure 5.1: True Input vs Predicted Output

Predicted vs. True Values

Figure 5.1 shows a scatter plot comparing the predicted turbulence values (R) from the linear regression model with the true values. The significant scatter of points away from the ideal 45-degree line (representing perfect agreement) further confirms the model's poor performance.

Comparing the Outputs

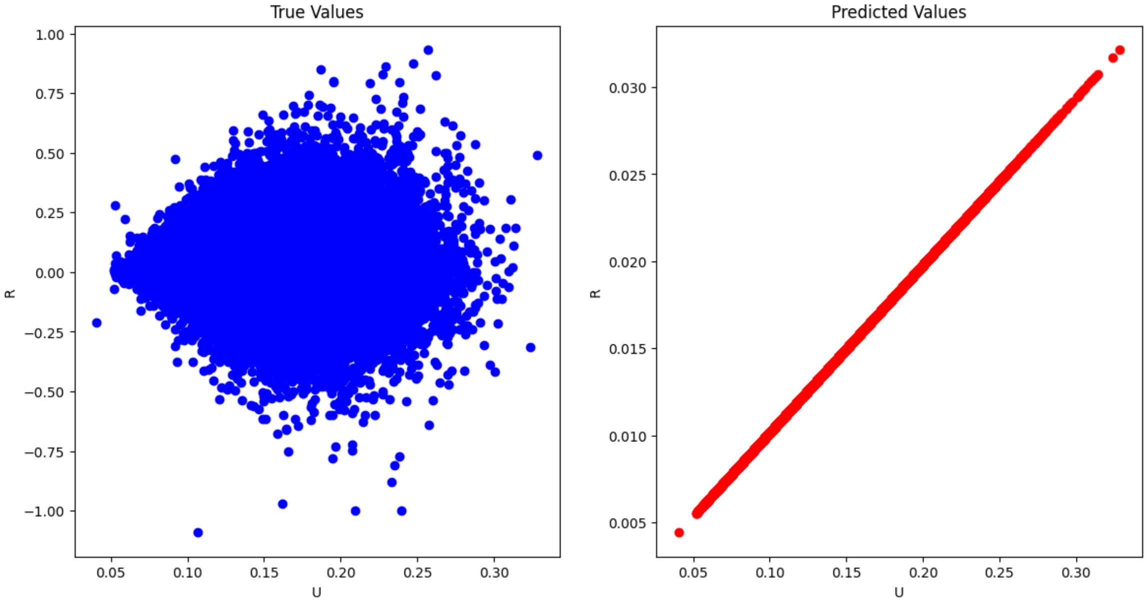

Figure 5.2: True Input vs Predicted Output

Figure 5.2 uses scatter plots to visualize the relationship between jet strength (U) and turbulence (R). The true output exhibits a complex, non-linear pattern. In contrast, the predicted output shows a simple linear trend, highlighting the linear regression model's inability to capture the complex relationships in the data.

Limitations

The linear regression model's poor performance underscores its limitations in modeling the complex dynamics of zonal jets. The low R^2 value, the scatter in the predicted vs. true values plot, and the failure to capture non-linear patterns in the color plots all point to the inadequacy of a linear model for this task.

Zonal jet dynamics are influenced by non-linear interactions between factors like turbulence, planetary rotation, and stratification. These complexities cannot be adequately represented by a simple linear model.

XGBoost Model Analysis

The XGBoost model was employed as another baseline model to investigate the relationship between jet strength (U) and turbulence (R) in zonal jet dynamics. XGBoost, a machine learning algorithm that combines multiple decision trees, is capable of capturing non-linear relationships in data.

Model Performance

The XGBoost model achieved an MSE of 0.0054 and an R^2 of 0.0013. The low R^2 value indicates that the model explains only a small fraction (0.13%) of the variance in turbulence (R), suggesting a poor fit to the data.

Metric	Value
MSE	0.0054
R^2	0.0013

Table 5.2: Performance metrics for the XGBoost model.

Predicted vs. True Values

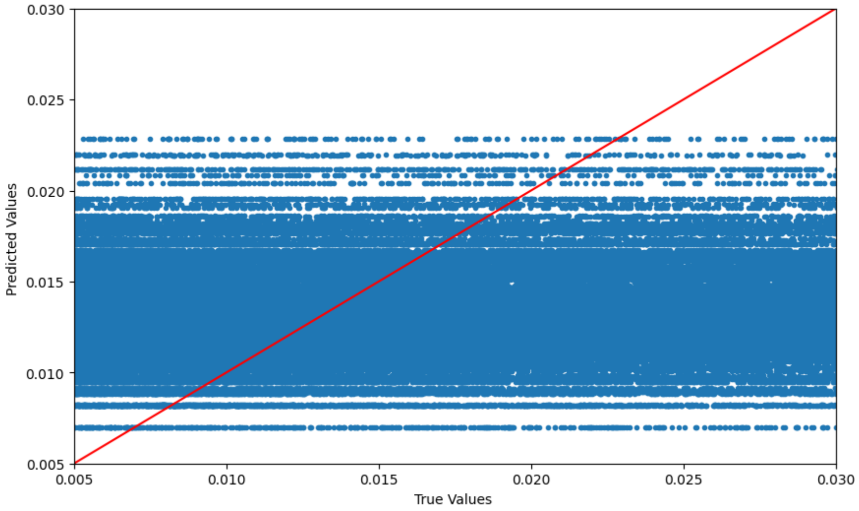

Figure 5.3: Predicted vs. True Values

Figure 5.3 shows a scatter plot comparing the XGBoost model's predicted turbulence values (R) with the true values. While some points cluster around the diagonal line, indicating agreement between predictions and true values, significant scatter suggests the model struggles to capture the complex dynamics accurately.

Comparing the Outputs

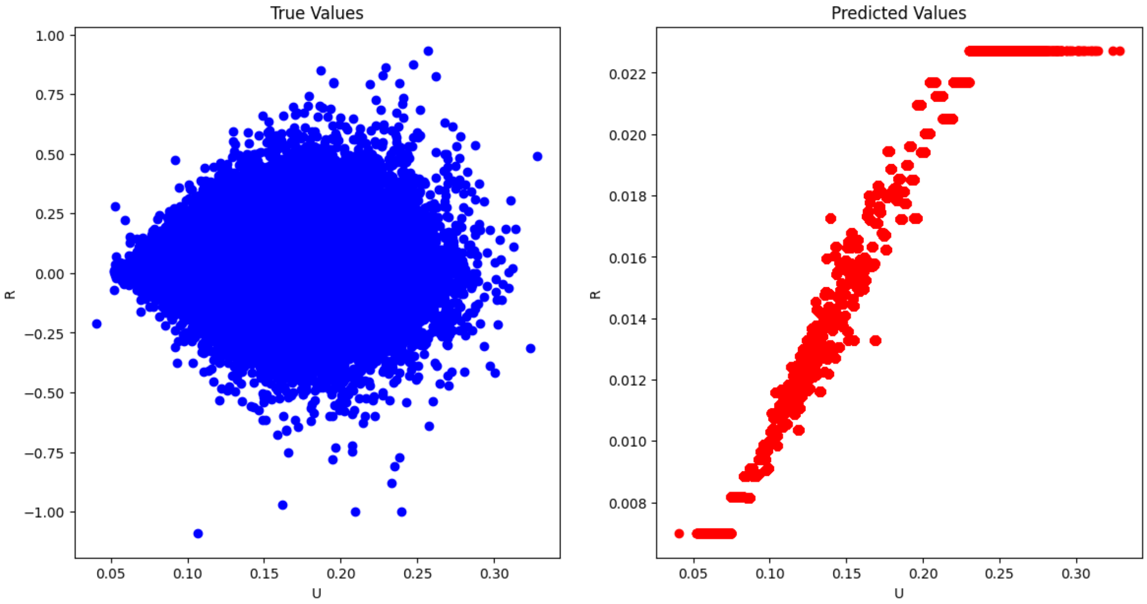

Figure 5.4: True Input vs Predicted Output

Figure 5.4 also uses scatter plots to visualize the relationship between jet strength (U) and turbulence (R). Much similar to the Linear Regression model, the true output here exhibits a complex, non-linear pattern, whereas, the predicted output shows a simple linear trend, also highlighting this model's inability to capture the complex relationships in the data.

Inability to Capture the Patterns

Both the linear regression and XGBoost models showed their limitations in accurately representing the relationship between jet strength (U) and turbulence (R). The linear regression model, restricted to linear relationships, failed to capture the inherent non-linearity of the system. XGBoost, despite its ability to model non-linearity, could not fully capture the patterns underlying the relationship due to its reliance on partitioning the input space.

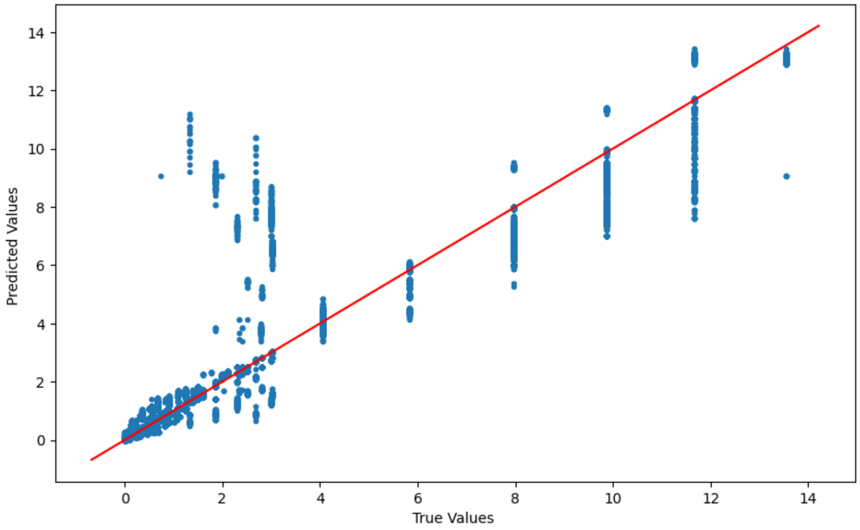

Figure 5.5: Predicted vs. True Values

5.1.2 XGBoost Model with Conditional Probability Data

Even though the XGBoost model was unable to learn the patterns in the relationship between jet strength (U) and turbulence (R), it was able to predict the conditional probability ($P(R|U)$) using jet strength (U) and turbulence (R) as input features.

Model Performance

The performance of this version XGBoost model was also evaluated using the MSE and R^2. This time, it achieved an MSE of 0.0669 and an R^2 of 0.9957, indicating excellent performance in predicting the conditional probability. The value further reinforces the model's exceptional performance, as it implies that the model can explain approximately 99.57 % of the variances in the conditional probability.

Metric	Value
MSE	0.0669
R^2	0.9957

Table 5.3: Performance metrics for the XGBoost model.

Predicted vs. True Values

Figure 5.5 shows a scatter plot comparing the true output values with the predicted values from this model. The data points cluster tightly around the 45-degree line, indicating a strong agreement between the predicted and true values. This strong

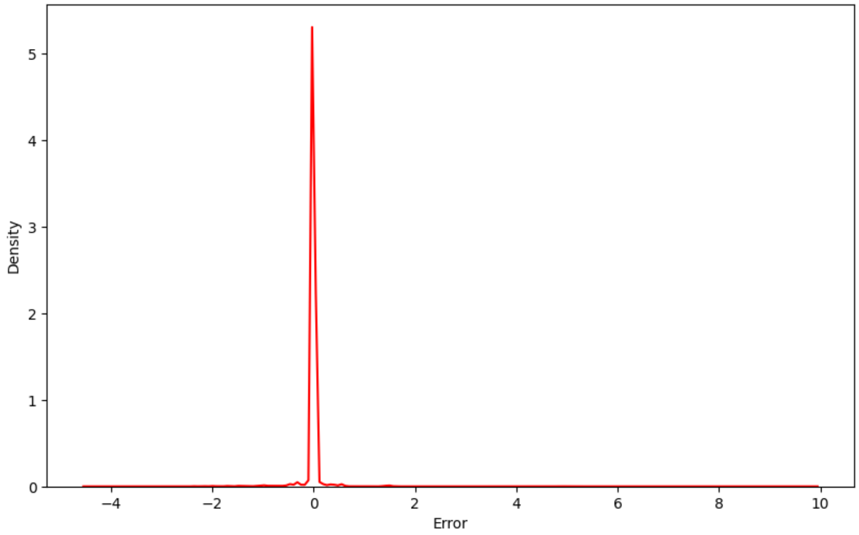

Figure 5.6: Prediction Error vs Density

linear relationship between the predicted and true values confirms the high R^2 value obtained, indicating that the model has successfully captured the underlying patterns and relationships in the data.

Error Distribution

Figure 5.6 displays the distribution of prediction errors using a kernel density estimation (KDE) plot. A KDE plot is a way to visualize the probability distribution of a dataset. The plot shows a unimodal distribution centered around zero, suggesting that the model's predictions are generally well-calibrated and have a low tendency for large errors.

True Output vs. Predicted Output

Figure 5.7 uses color-coded scatter plots to visualize the relationship between jet strength (U), turbulence (R), and the conditional probability ($P(R|U)$). The color scale represents the probability density, with warmer colors indicating higher densities and cooler colors indicating lower densities.

The color plot of the predicted output closely resembles the true output plot, demonstrating that the XGBoost model has successfully captured the complex patterns and distributions in the data.

Interpretation

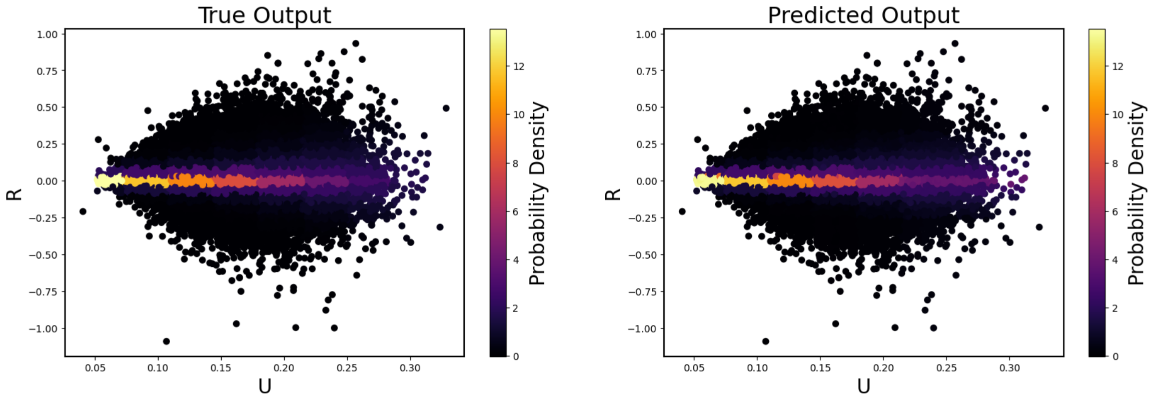

Figure 5.7: Color Plots to visualize the relationship between jet strength (U), turbulence (R), and the conditional probability ($P(R|U)$)

This version of the XGBoost model with a changed input and output data effectively captured the complex dynamics of zonal jets, as evidenced by the low MSE, high R^2 value, and the strong agreement between the predicted and true values. While demonstrating promising results in capturing the patterns of zonal jets, it also highlighted the need to explore more sophisticated modeling techniques. XGBoost, as a tree-based ensemble method, relies on partitioning the input space into distinct regions and making predictions based on simple rules within each region.

From Baseline Models to Deep Learning

The baseline models, linear regression and XGBoost, served as a crucial proof of concept, demonstrating the feasibility of using machine learning to model zonal jet dynamics while also showing their limitations which called the need for more sophisticated approaches. Linear regression, inherently limited to linear relationships, failed to capture the complex, non-linear dynamics evident in the data. While XGBoost, with its ability to model non-linearity, showed both promising and disappointing results. The ability of deep learning models to learn hierarchical representations of data, capture both spatial and temporal complexities, and process sequential information makes them well-suited for modeling the evolution of zonal jets over time and understand the underlying patterns.

5.2 Deep Learning Models

Feedforward Neural Network (FNN) Models

We analyze the performance of two Feedforward Neural Network (FNN) models in predicting zonal jet strength (R) based on zonal wind speed (U). The first model, referred to as the "Simple FNN," uses only zonal wind speed (U) as input. The second model, the "FNN with Features," incorporates additional features derived from U to enhance its predictive capabilities.

5.2.1 Simple FNN Model

Model Performance

The Simple FNN model's performance was evaluated using the Mean Squared Error (MSE) on a separate test dataset.

Metric	Value
Test MSE	0.0054

Table 5.4: Performance metrics for the Simple FNN model.

The model achieved a test MSE of 0.0054, suggesting initially promising performance in predicting zonal jet strength (R) based solely on zonal wind speed (U).

Predictions vs. True Values

Figure 5.8 presents a scatter plot comparing the Simple FNN model's predicted zonal jet strength (R) values with the true values. While the points cluster around the 45-degree line (representing perfect agreement), a closer examination reveals that the model struggles to capture the non-linear relationship between U and R accurately.

Training and Validation Loss

Figure 5.9 displays the training and validation loss curves during the training process for the Simple FNN. The steady decrease and convergence of both curves suggest that the model is learning effectively and generalizing well to unseen data. However, the relatively high MSE values indicate that the model's predictive accuracy is limited.

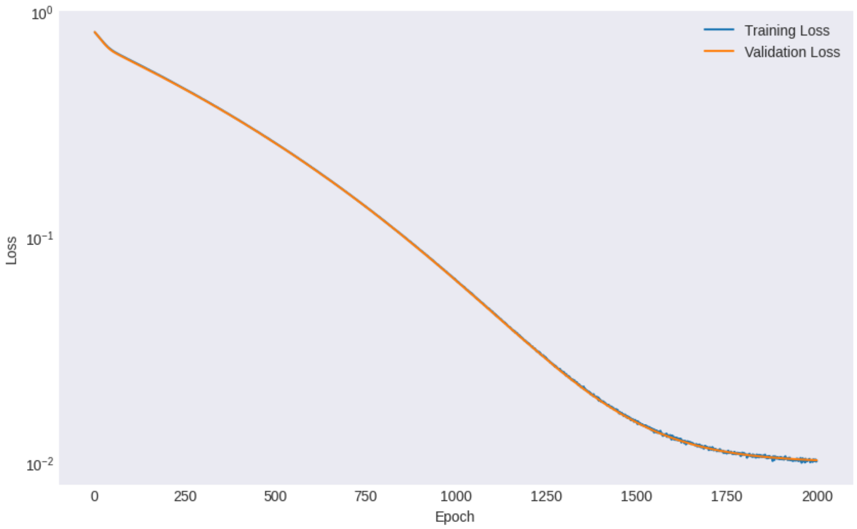

Figure 5.8: Scatter plot comparing the model's predicted turbulence values (R) with the true values. The red line represents perfect agreement.

True Output vs. Predicted Output

Figure 5.10 visually compares the true zonal jet strength (R) values (blue dots) with the Simple FNN model's predicted values (orange dots) against the input zonal wind speed (U). The plot reveals a distinct pattern in the true output, where higher R values occur at intermediate U values, indicating a complex, non-linear relationship. The Simple FNN model fails to capture this non-linearity effectively, as its predictions follow a more linear trend.

5.2.2 FNN Model with Features

To improve the model's ability to capture the complex relationship between zonal wind speed and zonal jet strength, additional features were incorporated into the FNN model. These features include:

- **U_autocorr:** Autocorrelation processed data from U, capturing temporal dependencies in the zonal wind speed.
- **mov_avg:** Moving average of U with a window size of 100, smoothing out short-term fluctuations.
- **mov_avg2:** Moving average of U with a window size of 1000, capturing longer-term trends.

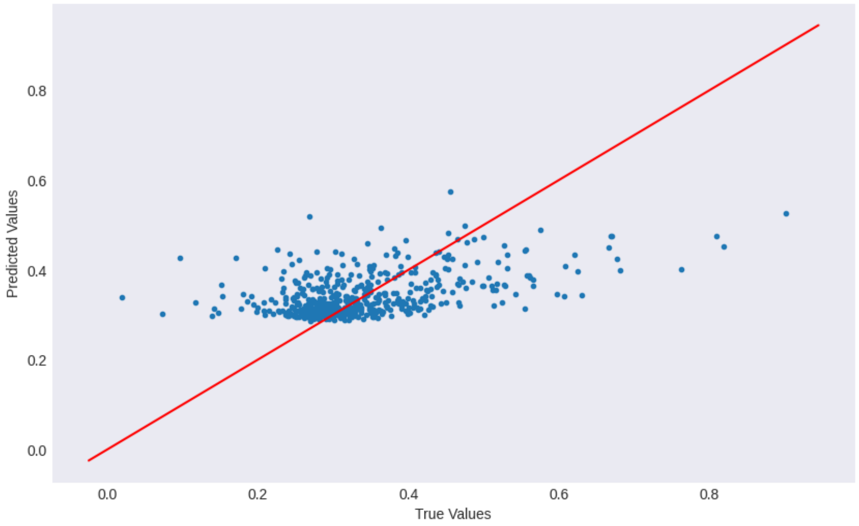

Figure 5.9: Training and validation loss of the model over training epochs. The blue line represents the training loss, and the orange line represents the validation loss.

- **mov_avg3:** Moving average of U with a window size of 10000, representing even longer-term variations.

Model Performance

The FNN with Features model achieved a test MSE of 0.0805, a significant improvement over the Simple FNN model.

Metric	Value
Test MSE	0.0805

Table 5.5: Performance metrics for the FNN with Features model.

Predictions vs. True Values

Figure 5.11 shows a scatter plot comparing the FNN with Features model's predictions for turbulence (R) with the true values. The points cluster more tightly around the diagonal line compared to the Simple FNN model, indicating improved predictive accuracy.

Training and Validation Loss

Figure 5.12 displays the training and validation loss of the FNN with Features model over training epochs. An epoch refers to one complete pass through the entire

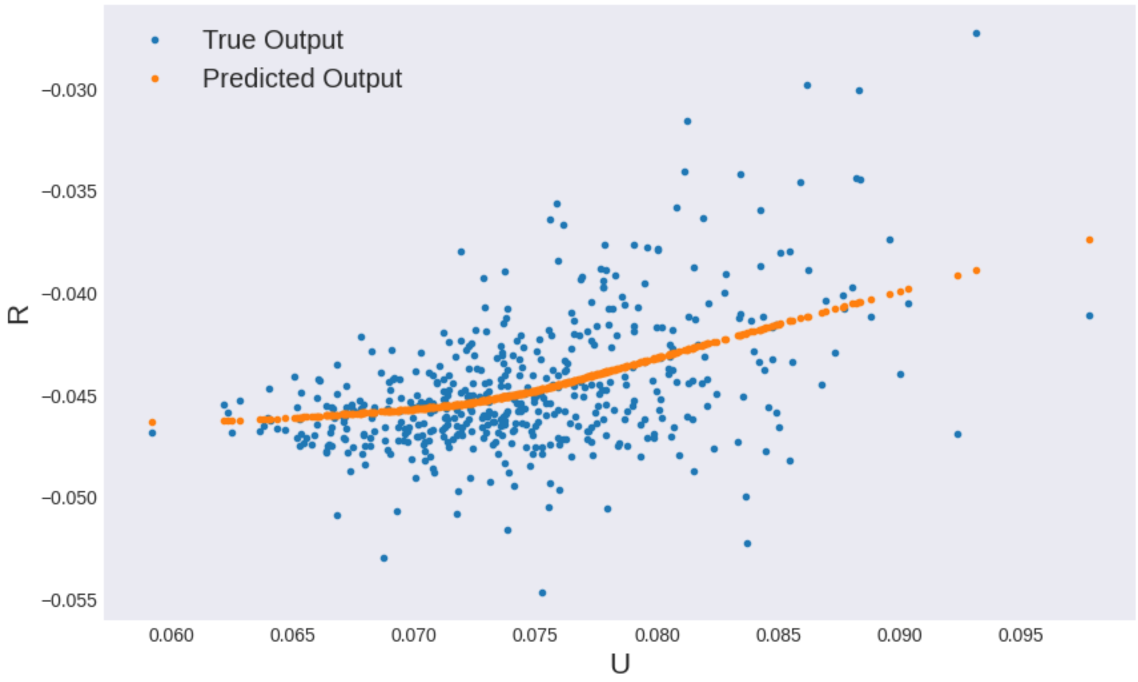

Figure 5.10: Scatter plots comparing the true output and the predicted output. The x-axis represents jet strength (U), and the y-axis represents turbulence (R)

training dataset. The decreasing training and validation loss, along with their convergence, suggest that the model is learning effectively and generalizing well to unseen data.

Color Plots: True Output vs. Predicted Output

Figure 5.13 shows scatter plots comparing the true output with the FNN with Features model's predicted output. The FNN with Features model demonstrates a marked improvement in capturing the complex, non-linear relationship between U and R, as evidenced by the closer resemblance of the predicted output to the true output. The smoother color gradient in the predicted output suggests that the model is producing continuous, probabilistic predictions, which is desirable for capturing real-world variability.

Discussion

The inclusion of additional features significantly enhanced the FNN model's ability to capture the complex relationship between zonal wind speed and zonal jet strength. The FNN with Features model achieved a lower test MSE and exhibited a better

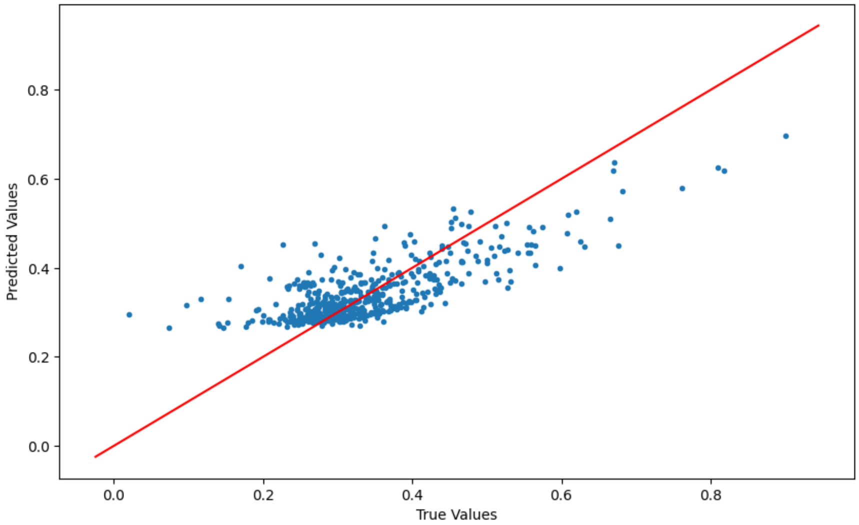

Figure 5.11: Scatter plot comparing the model's predicted turbulence values (R) with the true values. The red line represents perfect agreement.

visual agreement between predicted and true values compared to the Simple FNN model. This improvement highlights the importance of feature engineering in machine learning, where carefully selected features can provide the model with valuable information for making accurate predictions.

The results suggest that incorporating temporal dependencies and smoothing out short-term fluctuations in the zonal wind speed through features like autocorrelation and moving averages can significantly improve the model's ability to capture the underlying dynamics of zonal jets. This finding underscores the importance of considering the temporal evolution of the input variables when modeling complex geophysical phenomena.

Enhancing Performance with Custom Loss Functions

This section builds upon the previous analysis of simple FNN models by introducing a modified FNN model that incorporates a custom loss function. This custom loss function is designed to capture the specific statistical relationship between zonal wind speed (U) and turbulence (R), as represented by the conditional probability $P(R|U)$.

5.2.3 FNN with Custom Loss Function

Model Performance

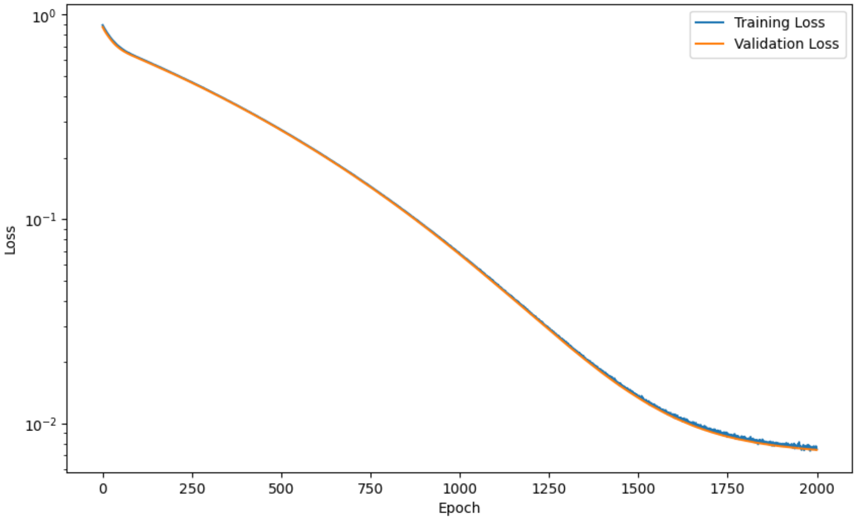

Figure 5.12: Training and validation loss of the model over training epochs. The blue line represents the training loss, and the orange line represents the validation loss.

The FNN with Custom Loss Function model was trained to predict the conditional probability $P(R|U)$ given U and R as inputs. The model's performance was evaluated using the MSE on a separate test dataset.

Metric	Value
Test MSE	0.0034

Table 5.6: Performance metrics for the FNN with Custom Loss Function model.

The model achieved a test MSE of 0.0034, a significant improvement over both the Simple FNN (MSE: 0.0054) and the FNN with Features (MSE: 0.0805) models. This lower MSE suggests that the custom loss function effectively guided the model towards learning the conditional probability distribution $P(R|U)$.

Predictions vs. True Values

Figure 5.14 shows a scatter plot comparing the FNN with Custom Loss Function model's predicted conditional probability values with the true values. The close alignment of the data points with the 45-degree line indicates a strong agreement between the predicted and true values, demonstrating the model's effectiveness in capturing the conditional probability distribution.

Training and Validation Loss

Figure 5.15 displays the training and validation loss curves during the training

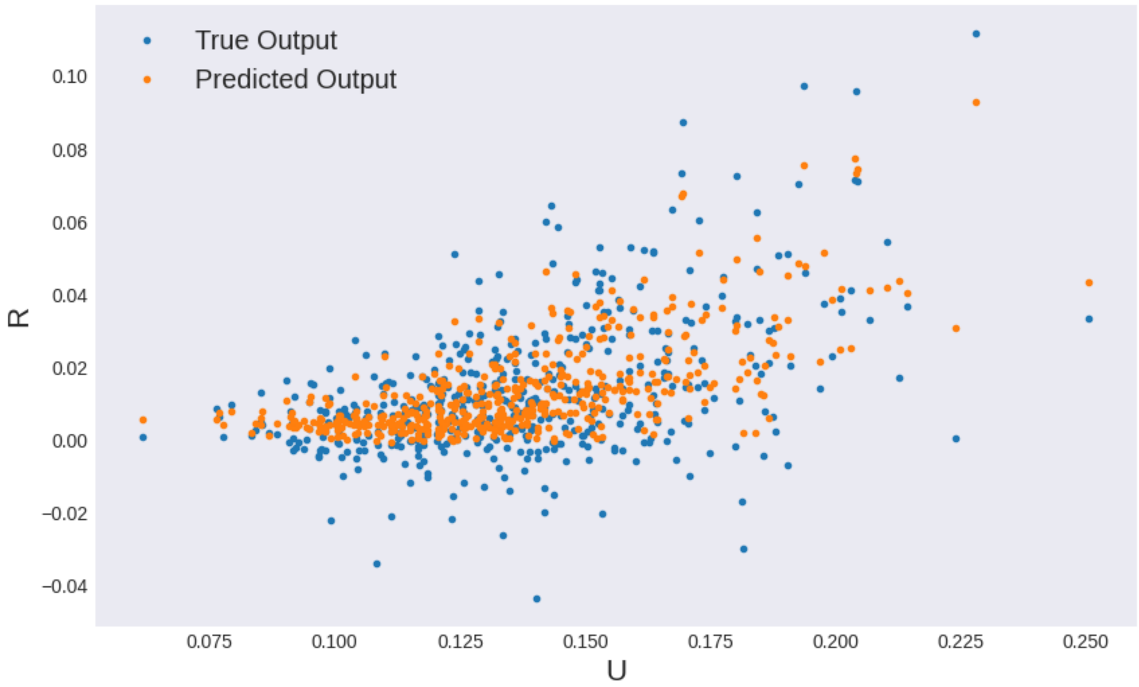

Figure 5.13: Scatter plots comparing the true output and the predicted output

process for the FNN with Custom Loss Function. The consistent decrease and convergence of both curves suggest that the model is learning effectively and generalizing well to unseen data.

Color Plots: True Output vs. Predicted Output

Figure 5.16 presents color plots comparing the true conditional probability distribution $P(R|U)$ with the predicted distribution from the FNN with Custom Loss Function model. The color scale represents probability density, with warmer colors indicating higher densities. The close resemblance between the true and predicted color plots highlights the model's ability to accurately capture the complex, nonlinear relationship between U and R as represented by the conditional probability distribution.

Discussion

The FNN with Custom Loss Function model demonstrated superior performance compared to the previous FNN models. This improvement can be attributed to the custom loss function, which explicitly guided the model towards learning the conditional probability distribution $P(R|U)$. By incorporating this statistical relationship

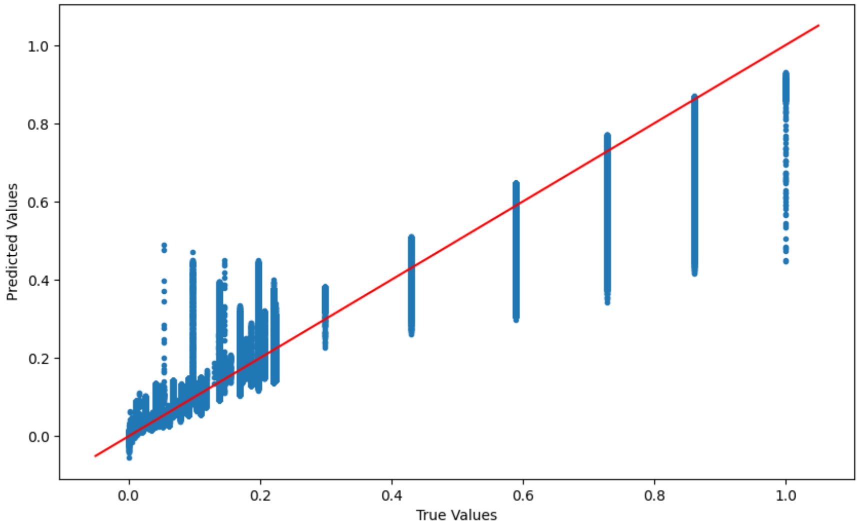

Figure 5.14: Scatter plot comparing the model's predicted turbulence values (R) with the true values. The red line represents perfect agreement.

into the loss function, the model was able to capture the dependencies between zonal wind speed and turbulence more effectively.

The results highlight the importance of tailoring the loss function to the specific characteristics of the problem being addressed. In this case, the custom loss function enabled the FNN model to learn a more accurate representation of the zonal jet dynamics, leading to improved predictions and a deeper understanding of the underlying physical mechanisms.

5.2.4 Statistics-Informed Neural Networks (SINNs)

Previous sections explored the capabilities of feedforward neural networks (FNNs) in modeling zonal jet dynamics. While FNNs showed promise, their limitations in capturing complex temporal dependencies and accurately representing the system's statistical properties motivated the exploration of a more sophisticated approach: Statistics-Informed Neural Networks (SINNs).

SINNs leverage the power of deep learning, specifically Long Short-Term Memory (LSTM) networks, to learn the underlying statistical properties of zonal jet systems directly from data. LSTMs are a type of recurrent neural network that excel at processing sequential information, making them well-suited for capturing the temporal

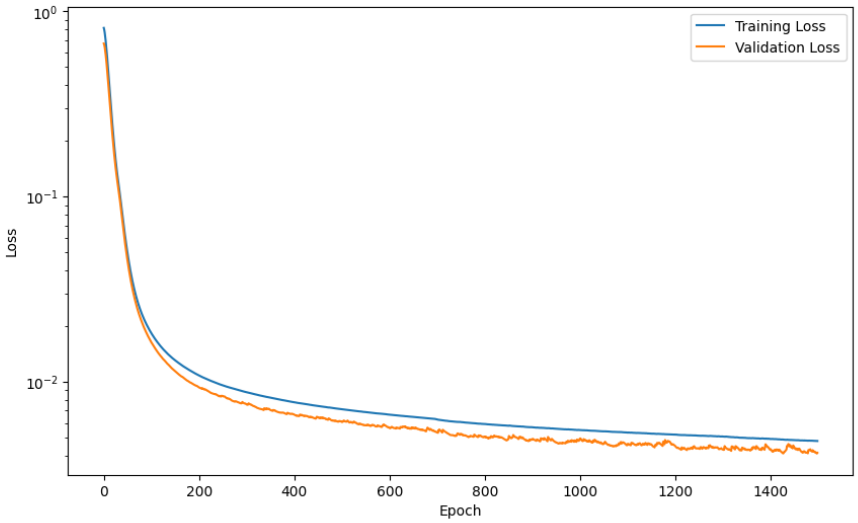

Figure 5.15: Training and validation loss of the model over training epochs. The blue line represents the training loss, and the orange line represents the validation loss.

evolution of zonal jets. Unlike FNNs, which focus on point-to-point predictions, SINNs aim to reproduce the statistical characteristics of the system, such as the probability density function (PDF) and the autocorrelation function (ACF).

Capturing the Variability of Zonal Jet Dynamics

Figure 5.17 illustrates the inherent variability and complexity of zonal jet dynamics by showing multiple realizations of the relationship between zonal wind speed (U) and zonal jet strength (R). Each colored dot represents a single realization, highlighting the non-linear and turbulent nature of the system. This complexity highlights the need for models capable of capturing the intricate patterns and statistical properties present in the data.

Target Empirical Statistics

Figure 5.18 presents the target empirical statistics that serve as the benchmark for the SINN model. These statistics include:

- **Autocorrelation Function (ACF):** The ACF (left panel) measures the correlation between the zonal jet strength at different time lags. The exponential decay observed in the ACF indicates the presence of memory effects and temporal correlations in the system.

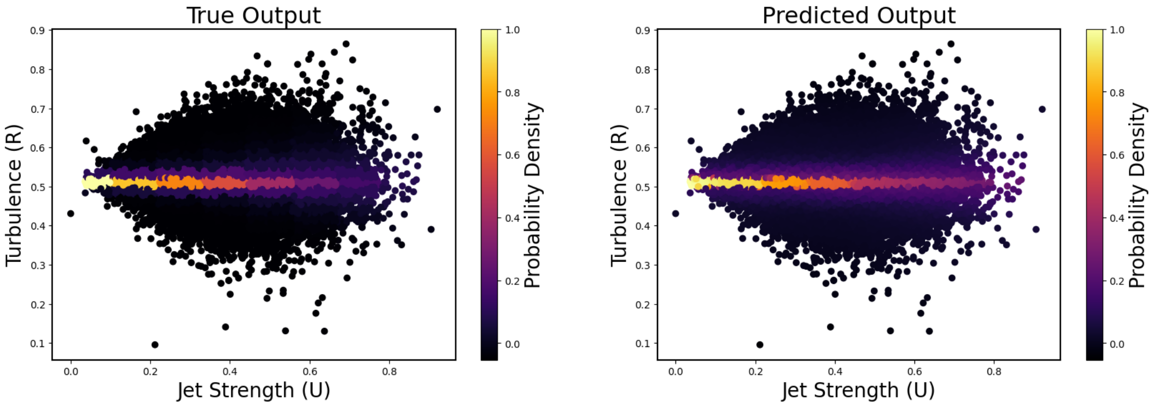

Figure 5.16: Scatter plots comparing the true output and the predicted output. The x-axis represents jet strength (U), and the y-axis represents turbulence (R)

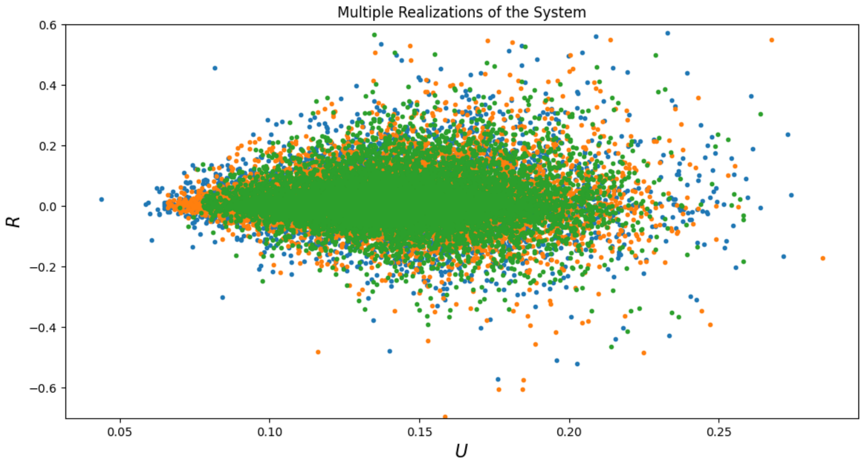

Figure 5.17: Multiple realizations of the interactions of zonal jets (U) and turbulence (R)

- **Probability Density Function (PDF):** The PDF (center panel) describes the probability distribution of zonal jet strength. The unimodal and symmetric distribution suggests an underlying stochastic process governing the dynamics.

- **ACF for Squared Target Variable (q^2):** The ACF for q^2 (right panel) provides insights into the higher-order statistical moments of the system.

The SINN model is trained to reproduce these target empirical statistics, ensuring that it accurately captures the statistical behavior of the zonal jet system.

Equilibrium Probability Density Function

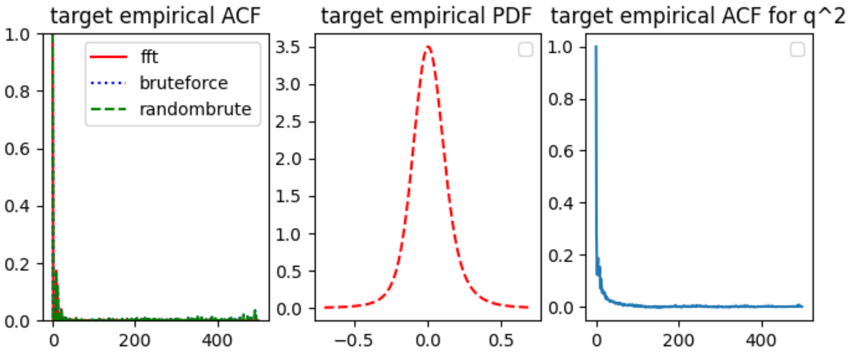

Figure 5.18: Some Target Empirical Statistics

Figure 5.19 compares the exact PDF of the target data (red line) with the PDF generated by the trained SINN model (green dashed line). The close alignment between the two PDFs demonstrates the SINN model's ability to accurately capture the equilibrium probability distribution of the zonal jet system. This accurate representation of the equilibrium PDF is crucial for understanding the long-term behavior and statistical properties of the system.

Trajectory Comparisons: Short-Term and Long-Term Dynamics

Figures 5.20 and 5.21 compare the trajectories generated by the SINN model (orange line) with those from a traditional Euler-Maruyama (EM) simulations (blue line). The close agreement between the trajectories, both in the short-term and long-term regimes, highlights the SINN model's ability to accurately capture the dynamics of the zonal jet system over a wide range of time scales. This capability is particularly valuable for applications like climate modeling, where accurate long-term predictions are essential.

Discussion

The SINN model demonstrated superior performance compared to the FNN models in capturing the complex dynamics and statistical properties of zonal jet systems. This success can be attributed to the use of LSTMs, which effectively handle temporal dependencies, and the focus on reproducing target empirical statistics, ensuring accurate representation of the system's statistical behavior.

The SINN approach offers a powerful and flexible framework for modeling complex geophysical phenomena. Its ability to learn directly from data, capture temporal

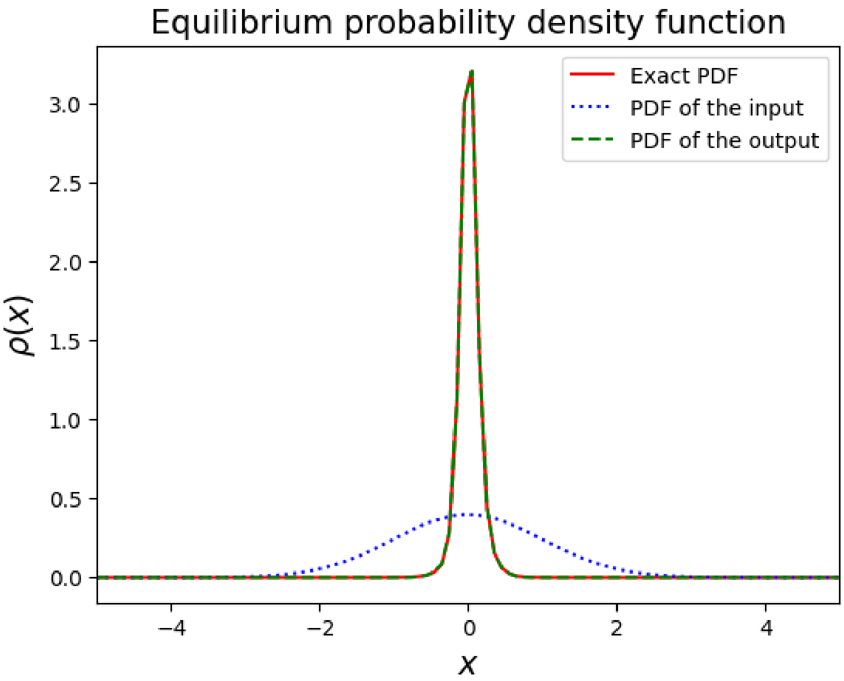

Figure 5.19: Comparison of the true PDF and predicted PDF

dependencies, and accurately represent statistical properties makes it a valuable tool for advancing our understanding of zonal jet dynamics and other complex systems.

5.2.5 Physics-Informed Neural Networks (PINNs)

Previous sections explored data-driven approaches, including FNNs and SINNs, for modeling zonal jet dynamics. While these models demonstrated the ability to learn from data, they did not explicitly incorporate the underlying physical laws governing the system. This section introduces Physics-Informed Neural Networks (PINNs), a novel approach that integrates deep learning with physical principles, enabling the discovery of governing equations and the calibration of unknown parameters directly from data.

Objective and Methodology

The objective of our PINN implementation is to leverage symbolic regression, a machine learning technique for discovering mathematical expressions, to uncover the underlying ordinary differential equations (ODEs) governing the zonal jet system.

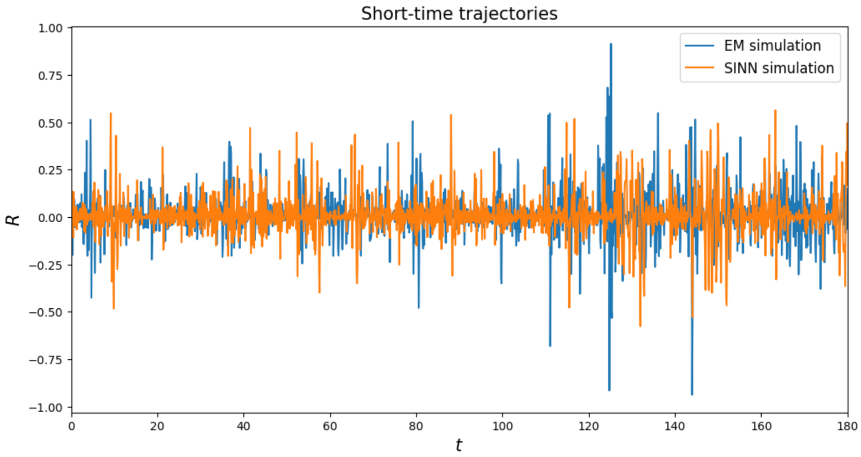

Figure 5.20: Comparison of the true PDF and predicted PDF

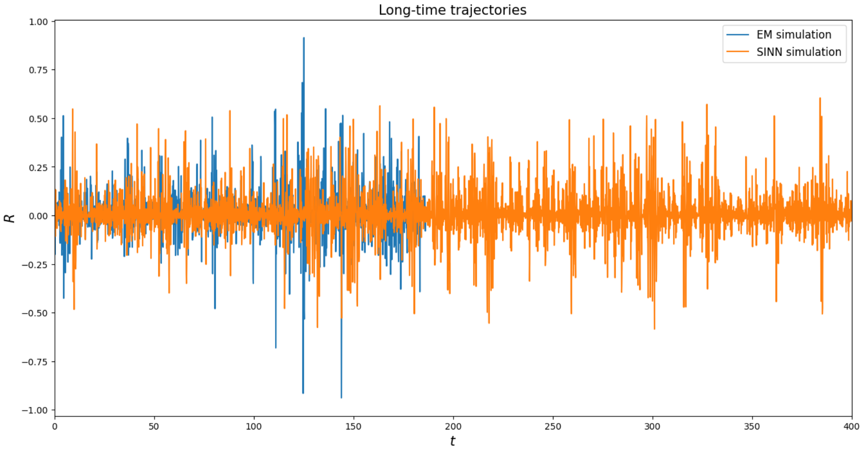

Figure 5.21: Comparison of the true PDF and predicted PDF

Specifically, we aim to discover the equations for the perturbation streamfunction (ψ_e) and the sheared wavenumber component (ψ_+), assuming knowledge of the ODE governing the zonal jet mean flow (U).

Our methodology involves:

1. **Generating Observational Data:** Using the Euler-Maruyama method, we generate data from a known ODE system, representing the time evolution of R (Reynolds stress divergence) and U (zonal jet mean flow). This data serves as the target for the PINN to learn.
2. **Training the PINN:** The PINN is trained using a combination of two loss functions:
   - **ODE Residual Loss:** Measures the discrepancy between the predicted and true values of the ODEs, ensuring the learned equations accurately describe the system's dynamics.
   - **Data Loss:** Quantifies the difference between the model's predictions and the observational data, ensuring the learned equations reproduce the observed behavior.
3. **Discovering Equations and Calibrating Parameters:** By minimizing the combined loss, the PINN learns to approximate the unknown ODEs for ψ_e and ψ_+ and calibrate any unknown parameters within these equations.

Observational Data Generation

Figure 5.22 shows the observational data generated from the known ODE system using the Euler-Maruyama method. This data, representing the time evolution of R and U, serves as the ground truth for the PINN to learn.

Training and Loss Functions

Figure 5.23 depicts the evolution of the ODE loss (left panel) and data loss (right panel) during the PINN training process. The fluctuating ODE loss indicates the model's efforts to minimize the residuals of the governing ODEs, while the decreasing data loss suggests the model is progressively improving its fit to the observational data.

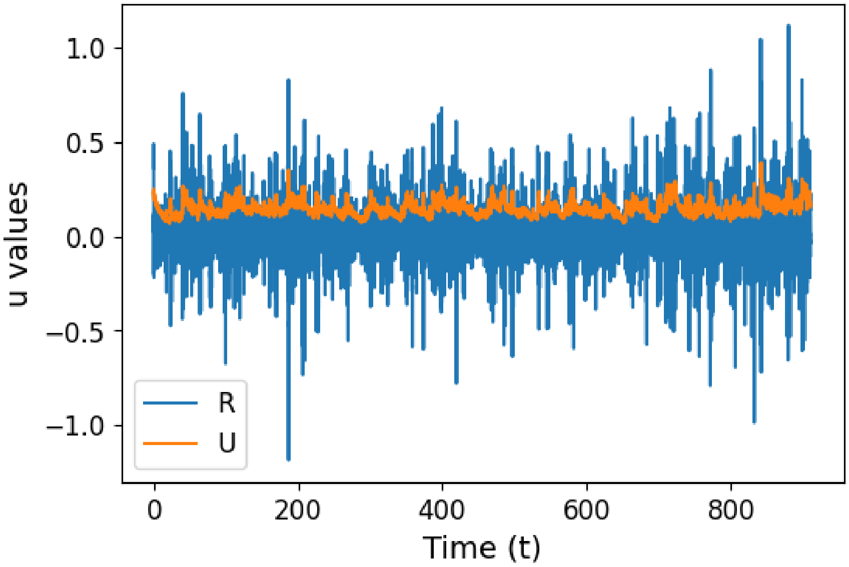

Figure 5.22: Simulation data of our ODE system

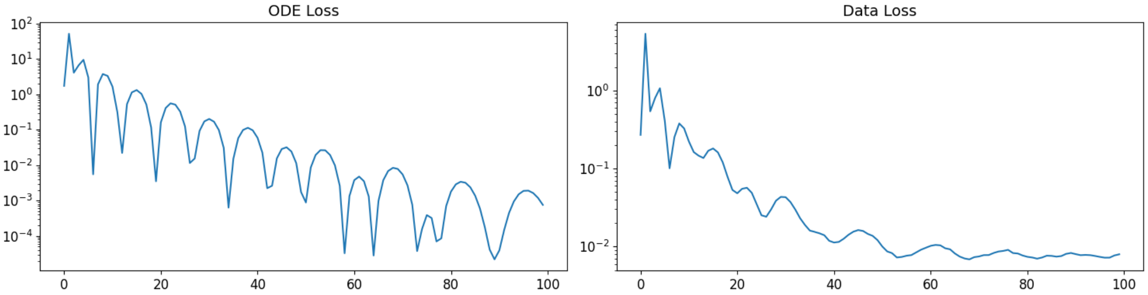

Figure 5.23: Simulation data of our ODE system

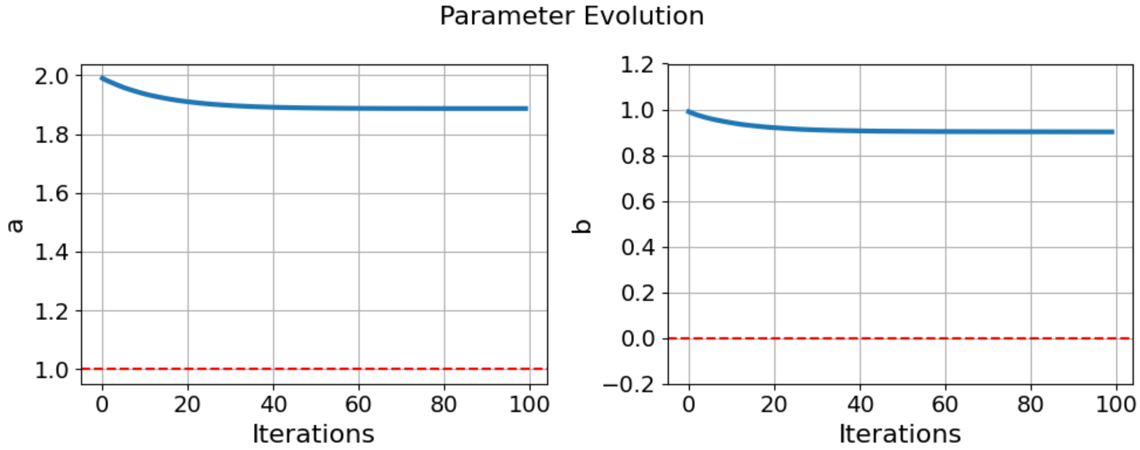

Figure 5.24: Evolution of the unknown parameters during the training process

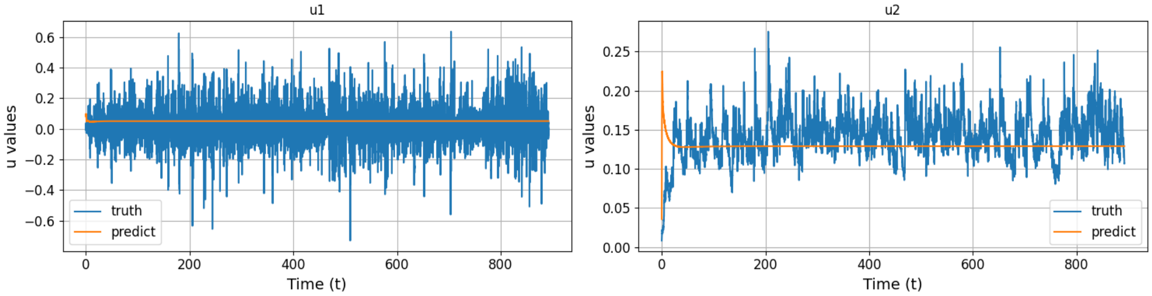

Figure 5.25: Comparison of Predictions with True Values

Parameter Evolution

Figure 5.24 illustrates the evolution of two unknown parameters, a and b, during the training process. The convergence of both parameters towards their true values (indicated by the red dashed lines) demonstrates the PINN's ability to calibrate unknown parameters within the governing equations directly from data.

Prediction Accuracy

Figure 5.25 compares the PINN's predictions for u_1 (related to ψ_e) and u_2 (related to ψ_+) with the true values. The close agreement, particularly in capturing the oscillatory behavior and overall trends, demonstrates the PINN's ability to accurately represent the complex dynamics of the zonal jet system.

Discussion

The PINN model successfully discovered the underlying ODEs governing the zonal jet system and calibrated unknown parameters directly from data. This achievement highlights the potential of PINNs in advancing our understanding of complex geophysical phenomena, particularly in scenarios where analytical solutions or explicit knowledge of the governing equations are lacking.

The integration of physical principles into the deep learning framework enables PINNs to provide physically consistent and interpretable predictions. This approach offers a promising route for closing the gap between data-driven and physics-based modeling.

5.3 Summary

This chapter presented a comprehensive analysis of various machine learning models applied to the study of zonal jet dynamics. Starting with baseline models like linear regression and XGBoost, we established the feasibility of using machine learning to predict zonal jet strength based on zonal wind speed. However, these models exhibited limitations in capturing the complex, non-linear, and turbulent nature of the system.

The exploration of deep learning models, specifically FNNs, demonstrated improved performance. The Simple FNN model, using only zonal wind speed as input, struggled to capture the non-linear relationship accurately. Incorporating additional features derived from zonal wind speed, such as autocorrelation and moving averages, significantly enhanced the model's predictive capabilities, as evidenced by the FNN with Features model. Further improvement was achieved by introducing a custom loss function tailored to capture the conditional probability distribution between zonal wind speed and turbulence, leading to the FNN with Custom Loss Function model, which outperformed all previous models.

Finally, the implementation of Statistics-Informed Neural Networks (SINNs) and Physics-Informed Neural Networks (PINNs) showcased the potential of integrating deep learning with statistical and physical constraints. SINNs, leveraging LSTM networks, successfully captured the temporal dependencies and statistical properties of the zonal jet system. PINNs, by incorporating physical laws into the learning process, demonstrated the ability to discover governing equations and calibrate unknown parameters directly from data.

These findings highlight the power and versatility of machine learning, particularly deep learning, in modeling complex geophysical phenomena like zonal jets. By incorporating statistical and physical constraints, deep learning models can provide accurate predictions, uncover hidden patterns, and offer valuable insights into the underlying physical mechanisms governing these systems. This research paves the way for further exploration of deep learning applications in zonal jet dynamics, including the development of more sophisticated models, the integration of multiple data sources, and the exploration of hybrid approaches that combine machine learning with traditional numerical simulations.

5.4 Future Directions

This research has laid a strong foundation for further exploration of zonal jet dynamics using deep learning. Several promising avenues for future research emerge from the findings presented in this thesis.

First, the analysis of the Physics-Informed Neural Network (PINN) model will be completed. This involves a more thorough investigation of the model's ability to discover the governing equations for ψ_e and ψ_+, including a detailed comparison of the discovered equations with the true equations used to generate the observational data. Additionally, the PINN's capability to calibrate unknown parameters will be further explored by testing its performance on datasets with varying levels of noise and missing data. This analysis will provide valuable insights into the robustness and generalizability of PINNs in uncovering the underlying physics of zonal jet systems.

Second, the research will dig deeper into the phenomenon of jet reversals, where the zonal jet undergoes a dramatic shift in direction. These reversals are associated with significant changes in weather patterns and can contribute to extreme weather events. By utilizing the deep learning models developed in this thesis, we aim to investigate the precursors and drivers of jet reversals, potentially leading to improved forecasting and mitigation strategies. The ability of deep learning models to capture complex temporal dependencies and non-linear interactions makes them particularly well-suited for studying these dynamic and often unpredictable events.

Finally, a more comprehensive comparison and analysis will be conducted between the deep learning models and the Statistical State Dynamics (SSD) model. This will involve using the SSD model as a reference for evaluating the deep learning models' performance and vice versa. By comparing the predictions, statistical properties, and physical interpretations derived from both approaches, we aim to identify areas where the models can complement each other and potentially lead to a unified framework that integrates data-driven learning with theoretical insights.

Bibliography

[1] C. L. Archer and K. Caldeira. Historical trends in the jet streams. *Geophysical Research Letters*, 35, 2008.

[2] J. M. Aurnou, M. Heimpel, L. Allen, E. M. King, and J. Wicht. Convective heat transfer and the pattern of thermal emission on the gas giants. *Geophysical Journal International*, 173:793–801, 2008.

[3] N. Bakas and P. Ioannou. Structural stability theory of two-dimensional fluid flow under stochastic forcing. *Journal of Fluid Mechanics*, 682:332–361, 2011.

[4] N. Bakas and P. Ioannou. Emergence of large scale structure in barotropic-plane turbulence. *Physical Review Letters*, 110, 2013.

[5] N. A. Bakas, N. C. Constantinou, and P. J. Ioannou. S3t stability of the homogeneous state of barotropic beta-plane turbulence. *Journal of the Atmospheric Sciences*, 72, 2015.

[6] N. A. Bakas and P. J. Ioannou. A theory for the emergence of coherent structures in beta-plane turbulence. *Journal of Fluid Mechanics*, 740, 2014.

[7] P. Baldi, P. Sadowski, and D. Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature Communications*, 5(1), July 2014.

[8] M. P. Baldwin, P. B. Rhines, H. P. Huang, and M. E. McIntyre. The jet-stream conundrum, 2007.

[9] S. Bastin, M. Claus, P. Brandt, and R. J. Greatbatch. Equatorial deep jets and their influence on the mean equatorial circulation in an idealized ocean model forced by intraseasonal momentum flux convergence. *Geophysical Research Letters*, 47, 2020.

[10] A. ben khalifa and H. Frigui. Multiple instance fuzzy inference neural networks. 10 2016.

[11] C. M. Bishop. *Pattern Recognition and Machine Learning*, volume 4 of *Information science and statistics*. Springer, 2006.

[12] L. Bottou. Large-scale machine learning with stochastic gradient descent. In Y. Lechevallier and G. Saporta, editors, *Proceedings of COMPSTAT'2010*, pages 177–186, Heidelberg, 2010. Physica-Verlag HD.

[13] G. Box, G. Jenkins, and G. Reinsel. *Time Series Analysis: Forecasting and Control*. Wiley Series in Probability and Statistics. Wiley, 2008.

[14] P. Brandt, A. Funk, V. Hormann, M. Dengler, R. Greatbatch, and J. Toole. Interannual atmospheric variability forced by the deep equatorial atlantic ocean. *Nature*, 473:497–500, 05 2011.

[15] S. L. Brunton, B. R. Noack, and P. Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52(Volume 52, 2020):477–508, 2020.

[16] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová. Machine learning and the physical sciences. *Rev. Mod. Phys.*, 91:045002, Dec 2019.

[17] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16. ACM, Aug. 2016.

[18] N. C. Constantinou, B. F. Farrell, and P. J. Ioannou. Emergence and equilibration of jets in beta-plane turbulence: Applications of stochastic structural stability theory. *Journal of the Atmospheric Sciences*, 71, 2014.

[19] G. V. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.

[20] D. G. Dritschel and M. E. McIntyre. Multiple jets as pv staircases: the phillips effect and the resilience of eddy-transport barriers. *Journal of the Atmospheric Sciences*, 65:855–874, 2008.

[21] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011.

[22] C. Eden and M. Dengler. Stacked jets in the deep equatorial atlantic ocean. *Journal of Geophysical Research: Oceans*, 113, 2008.

[23] Espa, S., Di Nitto, G., and Cenedese, A. The emergence of zonal jets in forced rotating shallow water turbulence: A laboratory study. *EPL*, 92(3):34006, 2010.

[24] B. Farrell and P. Ioannou. Statistical state dynamics: A new perspective on turbulence in shear flow. *Zonal Jets*, 2014.

[25] B. F. Farrell and P. J. Ioannou. Structural stability of turbulent jets. *Journal of the Atmospheric Sciences*, 60, 2003.

[26] E. Firing. Deep zonal currents in the central equatorial pacific. 1987.

[27] J. G. Fitzgerald and B. F. Farrell. Statistical state dynamics of vertically sheared horizontal flows in two-dimensional stratified turbulence. *Journal of Fluid Mechanics*, 854, 2018.

[28] B. Galperin, H. Nakano, H. Huang, and S. Sukoriansky. The ubiquitous zonal jets in the atmospheres of giant planets and earth's oceans. *Geophysical Research Letters*, 31, 2004.

[29] B. V. Gnedenko and O. B. Sheĭnin. *The Theory of Probability*, pages 211–282. Birkhäuser Basel, Basel, 1992.

[30] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning.* MIT Press, 2016. `http://www.deeplearningbook.org`.

[31] I. J. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning.* MIT Press, Cambridge, MA, USA, 2016. `http://www.deeplearningbook.org`.

[32] A. L. Gordon. Interocean exchange of thermocline water. *Journal of Geophysical Research: Oceans*, 91:5037–5046, 1986.

[33] A. Hall and M. Visbeck. Synchronous variability in the southern hemisphere atmosphere, sea ice, and ocean resulting from the annular mode. *Journal of Climate*, 15(21):3043 – 3057, 2002.

[34] Y. Ham, J.-H. Kim, and J. Luo. Deep learning for multi-year enso forecasts. *Nature*, 573:568 – 572, 2019.

[35] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin. The elements of statistical learning: Data mining, inference, and prediction. *Math. Intell.*, 27:83–85, 11 2004.

[36] G. Hinton, l. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29:82–97, 11 2012.

[37] J. Holton and G. Hakim. *An Introduction to Dynamic Meteorology.* Number v. 88 in An Introduction to Dynamic Meteorology. Elsevier Science, 2013.

[38] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

[39] H.-P. Huang, B. Galperin, and S. Sukoriansky. Anisotropic spectra in two-dimensional turbulence on the surface of a rotating sphere. *Physics of Fluids*, 13(1):225–240, 2001.

[40] A. P. Ingersoll. Atmospheric dynamics of the outer planets. In R. Pearce, editor, *Meteorology at the Millennium*, volume 83 of *International Geophysics*, pages 306–315. Academic Press, 2002.

[41] M. Ismail and R. Clarke. Adaptive block/location coding of facsimile signals using subsampling and interpolation for pre- and postprocessing. *IEEE Trans. Commun.*, 29:1925–1934, 1981.

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

[43] R. H. Kraichnan. Inertial ranges in two-dimensional turbulence. *Physics of fluids*, 10(7):1417, 1967.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[45] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–44, 05 2015.

[46] Y. LeCun, Y. Bengio, and G. E. Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[47] J. M. Lewis. Ooishi's observation: Viewed in the context of jet stream discovery. *Bulletin of the American Meteorological Society*, 84(3):357 – 370, 2003.

[48] V. Markapuri, G. LaVessi, R. Stewart, and D. Wagner. Bombus species image classification. 06 2020.

[49] J. Marshall, D. Ferreira, J. Campin, and D. Enderton. Mean climate and variability of the atmosphere and ocean on an aquaplanet. *Journal of the Atmospheric Sciences*, 64:4270–4286, 2007.

[50] M. L. Martín, S. L. Clainche, and B. Carro. Model-free short-term fluid dynamics estimator with a deep 3d-convolutional neural network. *Expert Syst. Appl.*, 177:114924, 2021.

[51] D. Maulud and A. Mohsin Abdulazeez. A review on linear regression comprehensive in machine learning. *Journal of Applied Science and Technology Trends*, 1:140–147, 12 2020.

[52] N. Maximenko, B. Bang, and H. Sasaki. Observational evidence of alternating zonal jets in the world ocean. *Geophysical Research Letters*, 32, 2005.

[53] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943.

[54] M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry.* MIT Press, Cambridge, MA, USA, 1969.

[55] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning, 2013.

[56] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress.

[57] D. Nualart. Kolmogorov and probability theory. *Arbor*, 178(704):607–619, ago. 2004.

[58] C. C. Porco, R. A. West, A. McEwen, A. D. D. Genio, A. P. Ingersoll, P. Thomas, S. Squyres, L. Dones, C. D. Murray, T. V. Johnson, J. A. Burns, A. Brahic, G. Neukum, J. Veverka, J. M. Barbara, T. Denk, M. Evans, J. J. Ferrier, P. Geissler, P. Helfenstein, T. Roatsch, H. Throop, M. Tiscareno, and A. R. Vasavada. Cassini imaging of jupiter's atmosphere, satellites, and rings. *Science*, 299(5612):1541–1547, 2003.

[59] M. Raissi, P. Perdikaris, and G. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, 378:686–707, 2019.

[60] P. Read, Y. Yamazaki, S. Lewis, P. Williams, K. Miki-Yamazaki, J. Sommeria, H. Didelle, and A. Fincham. Jupiter's and saturn's convectively driven banded jets in the laboratory. *Geophysical Research Letters*, 31, 2004.

[61] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, and M. Prabhat. Deep learning and process understanding for data-driven earth system science. *Nature*, 566:195, 02 2019.

[62] P. B. Rhines. Waves and turbulence on a beta-plane. *Journal of Fluid Mechanics*, 69(3):417–443, 1975.

[63] K. J. Richards, N. Maximenko, F. O. Bryan, and H. Sasaki. Zonal jets in the pacific ocean. *Geophysical Research Letters*, 33, 2006.

[64] S. R. Rintoul and S. Sokolov. Baroclinic transport variability of the antarctic circumpolar current south of australia (woce repeat section sr3). *Journal of Geophysical Research: Oceans*, 106(C2):2815–2832, 2001.

[65] H. Robbins and S. Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400 – 407, 1951.

[66] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958.

[67] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[68] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, Jan. 2015.

[69] A. Showman, A. Ingersoll, R. Achterberg, and Y. Kaspi. The global atmospheric circulation of saturn. *Saturn in the 21st Century*, 2018.

[70] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503, 2016.

[71] S. H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*. Westview Press, 2000.

[72] S. Sukoriansky, N. Dikovskaya, and B. Galperin. On the arrest of inverse energy cascade and the rhines scale. *Journal of the Atmospheric Sciences*, 64:3312–3327, 2006.

[73] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[74] B. Taguchi, R. Furue, N. Komori, A. Kuwano-Yoshida, M. Nonaka, H. Sasaki, and W. Ohfuchi. Deep oceanic zonal jets constrained by fine-scale wind stress curls in the south pacific ocean: a high-resolution coupled gcm study. *Geophysical Research Letters*, 39, 2012.

[75] A. Thompson and J. Sallée. Jets and topography: Jet transitions and the impact on transport in the antarctic circumpolar current. *Journal of Physical Oceanography*, 42:956–972, 2012.

[76] T. Tieleman. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, 2012.

[77] G. K. Vallis. Atmospheric and oceanic fluid dynamics. 2006.

[78] G. K. Vallis and M. E. Maltrud. Generation of mean flows and jets on a beta plane and over topography. *Journal of Physical Oceanography*, 23(7):1346 – 1362, 1993.

[79] A. Vasavada and A. Showman. Jovian atmospheric dynamics: an update aftergalileoandcassini. *Reports on Progress in Physics*, 68:1935–1996, 2005.

[80] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[81] G. P. Williams. Planetary circulations: 1. barotropic representation of jovian and terrestrial turbulence. *Journal of Atmospheric Sciences*, 35(8):1399 – 1426, 1978.

[82] P. D. Williams and C. W. Kelsall. The dynamics of baroclinic zonal jets*. *Journal of the Atmospheric Sciences*, 72:1137–1151, 2015.

[83] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization, 2017.

[84] Y. Zhang, D. Zhang, and H. Jiang. Review of challenges and opportunities in turbulence modeling: A comparative analysis of data-driven machine learning approaches. *Journal of Marine Science and Engineering*, 11(7), 2023.

[85] Y. Zhu, Y.-H. Tang, and C. Kim. Learning stochastic dynamics with statistics-informed neural network. *Journal of Computational Physics*, 474:111819, 2023.