Bachelor Thesis

Multiscale modeling of a closure problem using the scattering transform

Maria Prat Colomer

Advisor: Joan Bruna Estrach (NYU)

Tutor: Antonio Pascual Iserte (UPC)

May 2023

Bachelor's degree in Mathematics
Bachelor's degree in Informatics Engineering

Universitat Politècnica de Catalunya

Abstract

Climate models estimate the dynamics of the fluids on Earth by solving equations on finite grids. The physical processes below the resolution of climate models have statistical effects on the resolved scales. To account for them, models incorporate subgrid parametrizations. Devising effective parametrizations is crucial when producing accurate, long-term climate predictions. In this work, we address this challenge by using the scattering transform to study and model the closure problem in the scattering domain and we develop a generative model and a super-resolution method for turbulence fields.

Keywords: climate, closure problem, generative modeling, machine learning, scattering transform, signal processing, super-resolution, turbulence

MSC2020: 68T07, 76F65, 94A12

Resum

Els models climàtics estimen la dinàmica dels fluids a la Terra resolent equacions contínues en quadrícules finites. Els processos físics per sota de la resolució dels models climàtics tenen efectes estadístics a les escales resoltes. Per tenir-los en compte, els models incorporen parametritzacions. El desenvolupament de parametritzacions efectives és crucial per produir prediccions climàtiques precises a llarg termini. En aquest treball, abordem aquest problema utilitzant l'*scattering transform* per estudiar i modelar el problema de clausura en el domini d'*scattering*, i desenvolupem un model generatiu i un mètode de superresolució per a camps de turbulència.

Paraules clau: clima, problema de clausura, modelatge generatiu, aprenentatge automàtic, *scattering transform*, processament del senyal, superresolució, turbulència

MSC2020: 68T07, 76F65, 94A12

Resumen

Los modelos climáticos estiman la dinámica de los fluidos en la Tierra resolviendo ecuaciones en cuadrículas finitas. Los procesos físicos por debajo de la resolución de los modelos climáticos tienen efectos estadísticos en las escalas resueltas. Para tenerlos en cuenta, los modelos incorporan parametrizaciones. El desarrollo de parametrizaciones efectivas es crucial para producir predicciones climáticas precisas a largo plazo. En este trabajo, abordamos este problema utilizando la *scattering transform* para estudiar y modelar el problema de clausura en el dominio de *scattering*, y desarrollamos un modelo generativo y un método de superresolución para campos de turbulencia.

Palabras clave: clima, problema de clausura, modelado generativo, aprendizaje automático, *scattering transform*, procesamiento del señal, superresolución, turbulencia

MSC2020: 68T07, 76F65, 94A12

Acknowledgements

I would like to express my gratitude to Joan Bruna for giving me the opportunity to work on this project, and for his time and guidance during my time in New York.

I would also like to thank my friends and family for their unwavering support.

Contents

1 Introduction

Climate models play a vital role in understanding and predicting the Earth's climate system. One important aspect that these models need to consider is the representation of ocean mesoscale eddies, which are large swirling motions of water in the ocean. These eddies hold a significant amount of the ocean's kinetic energy and are crucial for transporting heat and trace chemicals. The choice of grid resolution in climate models is a critical decision. If the resolution is too fine, it can result in lengthy simulations. On the other hand, a coarse grid may fail to resolve many important processes. To address this challenge, climate scientists design subgrid parametrization techniques. These techniques involve incorporating statistical models of unresolved scales to achieve physically realistic results while keeping the simulations computationally efficient.

On another topic, the scattering transform is a translation-invariant representation of a signal obtained by cascading wavelet transforms, applying non-linearities, and averaging the wavelet coefficients. This transform is stable to additive noise and Lipschitz continuous to the action of small diffeomorphisms, while preserving the energy of the signal.

The objective of our work is to leverage the properties of the scattering transform to address the closure problem in large eddy simulations. To achieve this, we develop a generative model for turbulence fields and a super-resolution method that can enhance low-resolution fields to high-resolution ones. These components form the foundation for the development of a subgrid parametrization approach that utilizes the scattering transform. In our study, we focus on potential vorticity fields generated using a quasi-geostrophic model, which is a testbed model that effectively captures the dynamics of mesoscale eddies.

This report is organized as follows. In Chapter 2, we provide an introduction to climate models and describe the model used to simulate ocean turbulence. Chapter 3 addresses the closure problem in large eddy simulations and explores the challenges encountered by data-driven parametrizations

when tackling this problem. Chapter 4 introduces the scattering transform and examines its various mathematical properties. Chapter 5 presents our implementation of the scattering transform, building upon the Kymatio package while making improvements to the existing code. We also conduct an analysis of its correctness and performance. In Chapter 6, we outline the synthesis method for turbulence fields. Chapter 7 focuses on the super-resolution method and provides a comparison with current approaches to demonstrate its potential.

The results presented in this thesis will be expanded upon in the near future as they show a promising approach to turbulence closure modeling.

Climate Models 2

Climate models are complex simulations that provide valuable insights into the future of our planet's climate. Climate refers to the statistical properties of meteorological variables over extended periods, ranging from months to decades and beyond. Unlike short-term weather forecasting, climate models focus on predicting long-term climate patterns.

By simulating various scenarios, such as changes in greenhouse gas emissions, climate models enable us to anticipate how the climate will evolve in the future. However, the accuracy of these models depends on various factors, such as their complexity and the computing resources available. Climate models are computationally expensive and require significant time and resources to produce reliable results. Achieving high accuracy can be particularly challenging, and even with the use of powerful supercomputers, it can be cost-prohibitive.

2.1 Grid-Based Climate Dynamics

In order to simulate Earth's climate, climate models divide the Earth's surface into a grid, assigning prognostic variables to each grid cell and facilitating a transfer of information between neighboring cells. Numerical methods are then used to solve equations of fluid motion for these variables to approximate the dynamics of Earth's fluids. These equations typically consist of partial differential equations, such as the Navier-Stokes equations or simplified versions, along with thermodynamics and conservation laws.

Depending on the model's specific objective, the grid configuration can be either two-dimensional or three-dimensional and may cover particular regions or the entire planet. This decision impacts the grid's resolution and determines the number of cells that can be taken into account while maintaining computational feasibility.

Climate dynamics are modeled using both scalar variables, such as air temperature, pressure, density, and water vapor

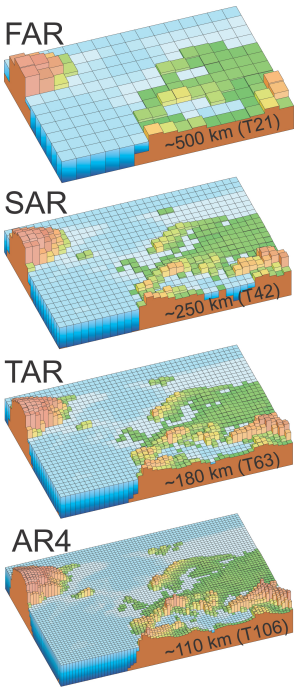


Figure 2.1: Spatial resolution evolution in global climate models featured in the IPCC Assessment Reports: FAR (IPCC, 1990), SAR (IPCC, 1996), TAR (IPCC, 2001a), and AR4 (2007). The figure shows the progressive improvement in horizontal resolution for Northern Europe across successive generations of climate models. Vertical resolution is not depicted, but has also increased in both atmospheric and oceanic models. Reprinted from (Le Treut, Somerville et al. 2007).

content, and vector variables, like wind. The physical laws governing these variables include the ideal gas law and conservation principles for air mass, water mass, energy, and air momentum. By solving these equations over space and time, we can simulate the behavior of Earth's climate. To represent the various components of Earth's climate, including the atmosphere, the ocean, and the ice, climate models are frequently subdivided and coupled together in global models (Eltahir and Krol 2021).

2.2 Multiscale Dynamics of the Ocean

The ocean serves as the planet's primary reservoir for heat and carbon. Ocean circulation redistributes these elements around the globe, having a major role in the evolution of the climate system. Ocean flow dynamics span a wide range of scales, from those imposed by land boundaries to Kolmogorov dissipation scales, which represent the smallest scales in turbulent flows and where kinetic energy dissipates into heat. This discussion will focus on three ocean processes occurring at different scales: ocean mixing, mesoscale ocean eddies, and large-scale ocean dynamics.

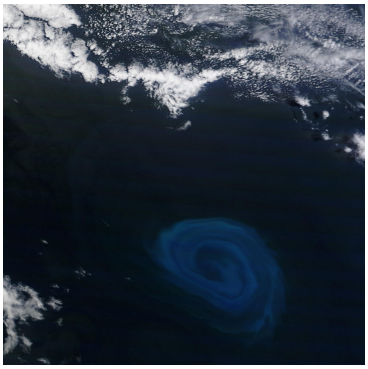


Figure 2.2: Anticyclonic 150-kilometer-wide eddy, located approximately 800 kilometers south of South Africa. Eddies within the Agulhas current are often among the world's largest. The light blue color is a result of a phytoplankton bloom. Reprinted from (Allen and NASA Earth Observatory 2011).

Ocean mixing, a small-scale process insufficiently represented in modern ocean models, has a considerable impact on the large-scale ocean climate. The mixing process is driven by mechanical forces (wind and tides), buoyancy forces in surface layers (ice formation), and friction forces at the ocean floor. This process alters the water's temperature, salinity, dissolved gases, and nutrient content, ultimately shaping oceanic conditions (Legg and Hallberg 2023).

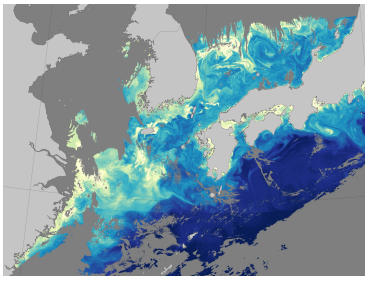


Figure 2.3: Chlorophyll concentrations in the waters off of South Korea and Japan showing the dynamics of surface currents. Reprinted from (Allen and NASA Earth Observatory 2016).

At larger scales, ocean mesoscale eddies are swirling strong water currents with horizontal scales ranging from tens to hundreds of kilometers and can persist for weeks to months (Adcroft, Griffies et al. 2023). Although their depth spans only a few kilometers, they account for the bulk of the kinetic energy in ocean circulation. Eddies play a crucial role in transporting heat, nutrients, and dissolved gases like carbon dioxide across vast distances. Figure 2.2 by (Allen and NASA Earth Observatory 2011) and Figure 2.3 by (Allen and NASA Earth Observatory 2016) depict eddies in the Agulhas Current and near Japan and South Korea,

respectively. These eddies upwell nutrients from the depths, sustaining marine life in surface waters.

Mesoscale eddies can form through two primary mechanisms: baroclinic instability and local topography. In the presence of baroclinic instability, eddies develop by drawing potential energy from the mean flow (Grotjahn 2003). Local topography can also create disturbances in the surrounding water, as observed in the eddy formed within the East Greenland Current (Smith, Morison et al. 1984).

Lastly, large-scale ocean dynamics significantly impact global and regional climate systems. One notable example is the Atlantic Meridional Overturning Circulation (AMOC), a circulation system that transports warm water northward and cold water southward, with both currents eventually looping back. It takes roughly a thousand years for a cubic meter of water to travel through the network of global ocean currents known as the global conveyor belt (NOAA 2023). The AMOC is an integral component of this global conveyor belt, affecting various climate phenomena such as Sahel and Indian summer monsoon rainfall, Atlantic hurricane activity, and Arctic climate conditions (Zhang 2023). Gaining insights into the dynamics of large-scale ocean circulation and associated low-frequency patterns is essential for accurately predicting not only low-frequency climate variability but also its overall contribution to the entire system.

2.3 Oceanic Dual Energy Cascade

Solar radiation, wind, and tides act on large scales of the order of 10^3 kilometers and are the primary external forces on the ocean. The energy acquired in these processes eventually dissipates as heat on much smaller scales, the Kolmogorov scales, which are around 1 millimiter in size. Oceanic turbulence redistributes the energy across this wide spectrum of scales, ranging from local ocean mixing processes to global ocean currents. The various scales interact in a nonlinear manner and relatively locally in frequency, so that the transfer occurs between scale-adjacent circulations.

A dual behavior can be observed in the ocean's turbulent energy cycle, as the kinetic energy flows in two directions.

On the one hand, eddies dissipate as their energy is transferred to smaller scales through a forward cascade of kinetic energy. On the other hand, there is also an inverse energy cascade, where energy from smaller scales is transferred back and powers large-scale circulations.

Within the mesoscale range, a small fraction of the energy dissipates towards the submesoscale range. However, the majority of the energy follows the inverse cascade, leading to the formation of large-scale structures in the ocean. For accurate predictions, simulations need to successfully capture the backscatter from small to large scales. If not, energy may accumulate and lead to numerically unstable and unrealistic models.

2.4 2-Layer Quasi-Geostrophic Model

The quasi-geostrophic equations are derived from simplifying the Navier-Stokes fluid equations of motion by considering hydrostatic and geostrophic balance. Hydrostatic balance refers to the balance between the pressure gradient and the gravitational force in the vertical direction, while geostrophic balance is the balance between the pressure gradient and the Coriolis force in the horizontal direction. These equilibriums are relevant in the context of climate modeling because large-scale processes in the atmosphere and the ocean are characterized by approximate hydrostatic and geostrophic equilibrium (Vallis 2017).

Quasi-geostrophic models have been proven useful to develop and test physics-based ocean parametrizations, as they can resolve ocean mesoscale eddies and be efficiently simulated in comparison to more sophisticated ocean models such as general circulation models (GCMs). GCMs are coupled atmosphere-ocean three-dimensional models that numerically solve the Navier-Stokes equations (Mechoso and Arakawa 2015).

As single-layer quasi-geostrophic models fail to describe oceanic vertical motions, we use multi-layer quasi-geostrophic models. In particular, a 2-layer idealized model is a common and suitable choice. The 2-layer quasi-geostrophic model considers two layers of homogeneous fluids with constant densities ρ_1 and ρ_2. The lighter fluid lies on top, ensuring the

system is gravitationally stable, with $\rho_1 \leq \rho_2$. The system is confined in a rotating box, and the interface between the two fluids is a horizontal plane. The depth of each layer is denoted by H_1 and H_2 (Pedlosky 1970).

The potential vorticity q_i and the streamfunction ψ_i of layer $i \in 1, 2$ are key variables in the model. The potential vorticity is modeled by quasi-geostrophic systems and is proportional to the scalar product of the absolute vorticity and the gradient of the potential temperature. The streamfunction ψ_i is related to the flow velocity components u_i and v_i in the x and the y direction, respectively, by the equations

$$u_i = -\frac{\partial \psi_i}{\partial y}, v_i = \frac{\partial \psi_i}{\partial x}.$$

Following (Abernathey, rochanotes et al. 2022), we write the 2-layer quasi-geostrophic evolution equations as

$$\partial_t q_1 + \mathrm{J}(\psi_1, q_1) + \beta_1 \partial_x \psi_1 = \mathrm{ssd} \tag{2.1}$$
$$\partial_t q_2 + \mathrm{J}(\psi_2, q_2) + \beta_2 \partial_x \psi_2 = \mathrm{ssd} - r_{ek} \nabla^2 \psi_2, \tag{2.2}$$

where the non-linear horizontal Jacobian is defined as

$$\mathrm{J}(u, v) = \partial_x u \partial_y v - \partial_y u \partial_x v.$$

The term r_{ek} denotes the Ekman friction parameter and ssd denotes small-scale dissipation. The mean potential vorticity gradients are $\beta_1 = \beta + F_1(U_1 - U_2)$ and $\beta_2 = \beta - F_2(U_1 - U_2)$, where the beta-plane approximation is used, such that the Coriolis acceleration is a linear function of latitude y with slope β, i.e. $f = f_0 + \beta y$. The quantity $U_1 - U_2$ is a fixed mean zonal velocity shear between the two fluid layers, $F_1 = \frac{k_d^2}{1+\delta}$ and $F_2 = \delta F_1$. The layer thickness ratio is $\delta = \frac{H_1}{H_2}$ and the deformation wavenumber is

$$k_d^2 = \frac{f_0^2}{g'} \frac{H_1 + H_2}{H_1 H_2},$$

where g' is the reduced gravity or buoyancy jump

$$g' = \frac{g(\rho_2 - \rho_1)}{\rho_1}$$

and f_0 is the Coriolis parameter. The ssd term accounts for the dissipation that occurs at small-scales when the energy

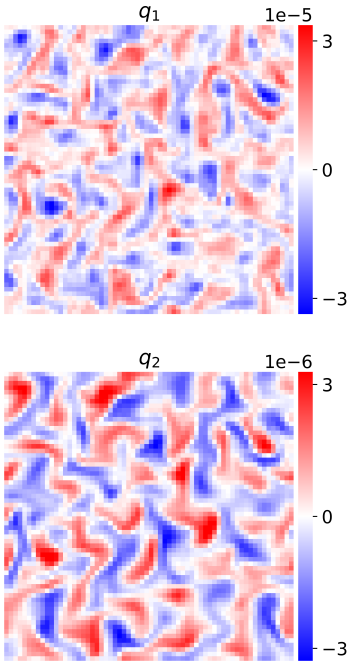


Figure 2.4: Potential vorticity fields of the top and bottom layers, generated using the PyQG package for the low-resolution configuration.

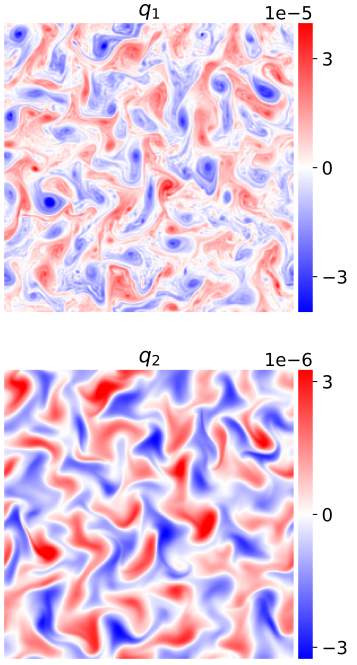


Figure 2.5: Potential vorticity fields of the top and bottom layers, generated using the PyQG package for the low-resolution configuration.

cascades down to them. It is implemented as an exponential filter that attenuates the energy in the small scales. The linear bottom drag in 2.2 dissipates kinetic energy from the large scales and smoothes the structures in the bottom layer. With this notation, the potential vorticities are

$$q_1 = \nabla^2\psi_1 + F_1(\psi_2 - \psi_1) \tag{2.3}$$
$$q_2 = \nabla^2\psi_2 + F_2(\psi_1 - \psi_2). \tag{2.4}$$

To simulate the 2-layer model we use PyQG, a Python package devised for simulating quasi-geostrophic dynamics in both atmospheric and oceanic fluid systems (Abernathey, rochanotes et al. 2022). It relies on a pseudo-spectral method that computes fast Fourier transforms considering periodic boundary conditions. We use the 2-layer model in the PyQG package because of its computational speed and capacity to resolve eddies, even within an idealized model framework. Although it does not capture all the details of complex climate models, PyQG effectively represents the essential fluid dynamics needed for our research and can lead to relevant insights applicable to other models. The model parameters mirror the eddy configuration outlined in (Ross, Li et al. 2023) and are given in Table 2.1 and Table 2.2.

Table 2.1: Parameters of the 2-layer quasi-geostrophic model used in the simulations. They correspond to the eddy configuration in (Ross, Li et al. 2023).

Simulation parameters		
Integration time		5 years
Time step	Δt	1 hour
Domain size	(L, W)	(1000, 1000) km
Ocean depth	$H = H_1 + H_2$	2500 m
Upper layer thickness	H_1	500 m
Bottom drag	r_{ek}	$5.787 \cdot 10^{-7}$ s^{-1}
Differential rotation	β	$1.5 \cdot 10^{-11}$ (m $\cdot$ s)$^{-1}$
Deformation radius	$r_d = \frac{1}{k_d}$	15 km
Mean flow	(U_1, U_2)	$(0.025, 0)$ m / s

Table 2.2: Grid parameters used in the simulations.

Grid parameters	Resolution	Grid step
High resolution	256×256	3.9 km
Low resolution	64×64	15.6 km

We will simulate 5-year trajectories on a 1000 km $\times$ 1000 km rectangular domain with a temporal resolution Δt of 1

hour. Our study considers two different spatial resolutions. The high-resolution configuration uses a 256×256 grid, translating to a grid step of 3.9 km, while the low-resolution configuration uses a 64×64 grid with a grid step of 15.6 km. Figure 2.4 and Figure 2.5 depict how the potential vorticity fields look like for these settings. We refer to the high-resolution simulation as eddy-resolving because it has the capacity to accurately resolve mesoscale eddies. On the other hand, the low-resolution configuration is termed eddy-permitting, given that mesoscale eddies can appear in the simulations, but their dynamics are not fully resolved (Chen, Barham et al. 2018). The simulations start with banded random initial conditions and go through a transient spin-up state until they reach the quasi-steady state after a few simulation years.

The Closure Problem of Large Eddy Simulations

3

Oceanic turbulence flows exhibit temporal and spatial scales of motion spanning many orders of magnitude, from the largest scales defined by the boundaries to the Kolmogorov scales where the disspation occurs, making direct simulations at high resolutions cost-prohibitive. Instead, large eddy simulations choose to only resolve low-frequency components of the spectrum and statistically model the effect of high-frequency components on the resolved scales. This approach involves fixing a resolution that is computationally feasible, but excludes processes occurring on smaller scales, such as ocean mixing, which can have an impact on larger scales due to the inverse energy cascade.

These overlooked processes are referred to as subgrid processes and can be modeled in climate models using subgrid parametrizations. Parametrizations model the statistical effects of subgrid processes on resolved scales, providing a computationally efficient way to achieve more realistic simulations without increasing the resolution. The ultimate goal is to design a closed system that accurately describes climate evolution, with a closure map that accounts for the impact of unresolved processes on modeled variables. Developing effective and comprehensive parameterizations remains one of the most critical challenges in creating accurate climate models and is currently an active area of research.

The main references for this chapter are (Lesieur 1997) and (Sagaut 2001).

3.1 Turbulence Closure Modeling

J. Boussinesq first introduced the problem of turbulence closure modeling, which has been extensively studied for over a century. Despite the long history of research in this area, it remains an active topic due to its importance and relevance to a wide range of fields.

3.1.1 Problem Statement

Let $q(\boldsymbol{x}, t)$ be the solution of the continuous equations we want to solve and let the dynamics of this field be described by the differential equation

$$\frac{\partial q}{\partial t} = F(q), \tag{3.1}$$

where F is a non-linear operator. When we are numerically solving this problem, we consider $\overline{q}$, the solution of the discretized equations on a grid. The problem then becomes

$$\frac{\partial \overline{q}}{\partial t} = \overline{F}(\overline{q}), \tag{3.2}$$

where the time-stepping is carried out with a numerical method, hence the introduction of the overline in $\overline{F}$. The filtered and coarse-grained field is denoted with $\overline{q}$. As the high-frequencies are attenuated, the field is typically subsampled onto a coarser grid after the filtering. It is important to note that the mapping between the high-resolution and low-resolutions representations is many-to-one.

The filtering process can be expressed as

$$q(\boldsymbol{x}, t) = \overline{q}(\boldsymbol{x}, t) + q'(\boldsymbol{x}, t) \tag{3.3}$$

$$\overline{q}(\boldsymbol{x}, t) = G * q(\boldsymbol{x}, t), \tag{3.4}$$

where G represents a low-pass filter and $q'(\boldsymbol{x}, t)$ represents the unresolved term. A detailed examination of the filtering process will be presented later. We are studying the explicit large eddy simulation approach as described in (Sagaut 2001), which involves adding a forcing term, known as the subgrid model, at each time-step when numerically solving the governing equations. The subgrid model is formulated in terms of the resolved scales, and several definitions of the subgrid forcing are commonly used in this context. We have chosen to utilize the total tendency definition given by

Note that we do an abuse of notation because we use $\overline{q}$ to denote indistintively the filtering or the filtering and coarse-graining of the field.

$$S = \overline{\frac{\partial q}{\partial t}} - \frac{\partial \overline{q}}{\partial t}, \tag{3.5}$$

which allows us to adjust the method at each step using

$$\frac{\partial \overline{q}}{\partial t} = \overline{F}(\overline{q}) + S. \tag{3.6}$$

In this scenario, the subgrid forcing accounts for the commutation error $\overline{\frac{\partial q}{\partial t}} \neq \frac{\partial \overline{q}}{\partial t}$ and includes the nonlinear advection present in the governing equations along with the numerical dissipation. Although this definition is less physically inspired than other alternatives, it is suitable for our purposes, as we aim to model S using a data-driven oracle. Moreover, tests have indicated that different definitions yield numerically equivalent results for the problem at hand.

A common strategy for accounting for the effects of unresolved processes in simulations is to run both high-resolution and low-resolution simulations with the same initial conditions. At each step of the low-resolution simulation, a subgrid correction term is added using Equation 3.6 to account for the unaccounted effects of unresolved scales. Hence, the subgrid term is problem-specific and depends heavily on the filter used for scale separation. Since it is incorporated into the numerical simulation workflow, it must also be computationally efficient and locally specific in space and time, while preserving the stability of the numerical simulation.

3.1.2 Scale-Separation Filtering

The scales in the solution are divided into resolved scales and subgrid scales, as depicted in Figure 3.1. The lower frequencies are computed spatially using a numerical method, while the higher frequencies are modeled through the subgrid term. In other words, the governing equations are solved on a coarser grid where small scales are not represented. The explicit separation between small and large scales is done with a low-pass filter.

In our case, we use a sharp filter such as the operator 1 in (Ross, Li et al. 2023), which attenuates the high-frequencies. Specifically, we consider the filter G with transfer function

$$\hat{G}(\boldsymbol{\omega}) = \begin{cases} 1 & \text{if } |\boldsymbol{\omega}| < \kappa \\ \exp[-23.6 \cdot (|\boldsymbol{\omega}| - \kappa)^4 \cdot (\Delta x)^4] & \text{otherwise,} \end{cases} \tag{3.7}$$

where Δx represents the spatial resolution of the coarser grid and κ is $\frac{2}{3}$ of the Nyquist frequency of the coarser grid.

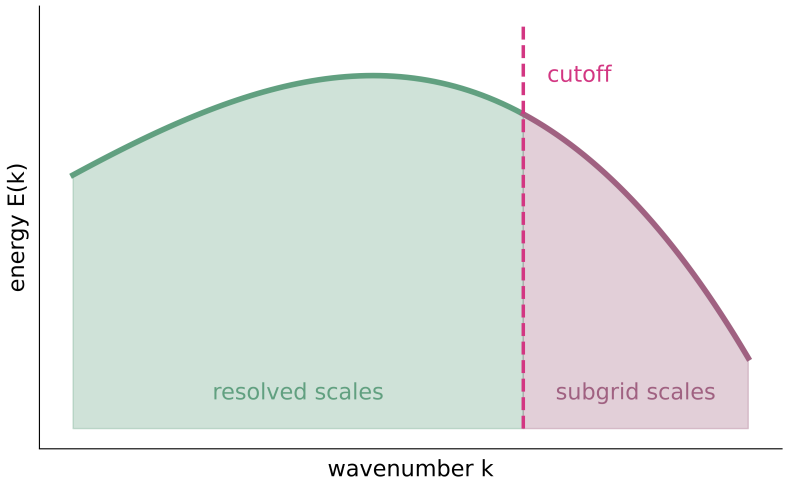


Figure 3.1: Illustration of the concept of scale separation in large eddy simulations. The x-axis represents the wavenumber k in the spectral domain, while the y-axis represents the energy of the field. The cutoff line depicted in the figure indicates the separation that would be obtained by a sharp cutoff filter. This filter separates the field into resolved and unresolved scales, with all frequencies below the cutoff line considered resolved and all those above the cutoff line considered unresolved.

By the definition of the convolution, the filter is linear,

$$G * (u + v) = G * u + G * v, \tag{3.8}$$

and it commutes with the time derivative

$$G * \frac{\partial q}{\partial t} = \frac{\partial}{\partial t}(G * q). \tag{3.9}$$

The filter G is not a Reynolds operator, which means that the filtered variable $\overline{q}$ is not equal to the double filtering of q, so $\overline{\overline{q}} \neq \overline{q}$, and that the filtering of the unresolved term $\overline{q'}$ does not vanish, so $\overline{q'} \neq 0$. In fact, this filter is invertible, since it has not yet projected the field onto a lower dimensional space. The resulting filtered field $\overline{q}$ is more regular than the original field q, as it attenuates the high frequencies.

This filter is non-local in space and non-local in frequency, and there is a little overlap in frequencies between the filtered field $\overline{q}$ and the unresolved term q'. However, it is close to a sharp cutoff filter, which is a common filtering choice that is non-local in space but local in frequency. Other classical filters include the top-hat filter, which is local in space but non-local in frequency, and the Gaussian filter, which is non-local both in frequency and space due to its non-compact support.

After the filtering, the data is subsampled in the spatial domain to obtain the desired resolution for the coarse-grained field. The cutoff frequency is chosen to retain all active scales in the filtered field, and this is the step where the loss of information occurs. We use this filter because it is the same filter utilized by PyQG implement the small-scale

dissipation in the governing equations. As a result, it is the most conservative choice of filter, given that the numerical method already applies it.

3.2 Challenges of Data-driven Parametrizations

Traditional approaches use fluid mechanics knowledge to identify certain properties in the resolved scales that imply the presence of small-scale effects, or they introduce latent variables like the kinetic energy of the subgrid scales. The modeling of large eddy simulations has been transformed by the use of data-driven parameterizations and machine learning techniques. For instance, Figure 3.2 by (Yuval and O'Gorman 2020) shows how a random forest parametrization significantly improves the accuracy of the modeling. Deep neural networks are a promising approach, as they can approximate arbitrary functions with a small number of parameters that does not scale exponentially with the problem space dimension. This makes them suitable for problems in high-dimensional spaces, such as the closure map we aim to model. However, these models are heavily reliant on high-quality training data, which is not always available. Moreover, it has been seen that neural networks can be unstable and may not generalize well to unseen regimes (Ross, Li et al. 2023).

To address these limitations, neural networks that incorporate physical priors into the model have been developed. These networks have shown improved accuracy and stability, even when training data is limited. For example, the incorporation of rotational equivariance priors and global enstrophy constraints in the loss function of a convolutional neural network has been shown to enhance the accuracy and stability of simulations (Guan, Subel et al. 2023). By reducing the space of possible functions, these added constraints guide the optimization towards more physically meaningful parameterizations.

Another recent development is the coupling of PDE solvers with automatic-differentiation toolchains (Rackauckas, Ma et al. 2020), which allows for the training of subgrid forcing terms embedded in the model's governing equations (Shankar, Puri et al. 2023). However, since most climate models are not differentiable, there remains a need to develop

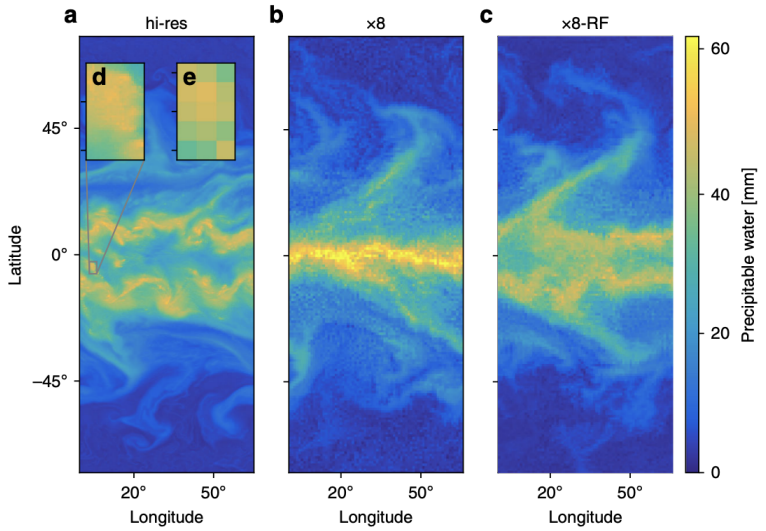


Figure 3.2: **a** shows a high-resolution simulation, **b** shows a low-resolution simulation and **c** shows a low-resolution simulation with a random forest parametrization. Reprinted from (Yuval and O'Gorman 2020).

parametrizations for subgrid terms that correct the numerical time-stepping, as explained in the previous section. These subgrid models can also provide valuable guidance for the development of new differentiable climate models and existing climate models can be run in ensemble to reinforce their results and provide more robust predictions. As such, they remain a key research focus in the field of climate modeling.

To explore the potential of the scattering transform for modeling subgrid forcings, we will use the 2-layer quasi-geostrophic model as a test bed. The scattering transform is particularly well-suited for this task due to its geometric properties and interpretability, which provide insights into the modeled physical processes.

The Scattering Transform 4

The scattering transform is a wavelet transform first introduced by S. Mallat in (S. Mallat 2012). It is built by recursively applying wavelet modulus decompositions followed by a lowpass filter. The resulting signal representation is both translation invariant and stable to the action of small diffeomorphisms. While Mallat's work laid the foundation for the scattering transform, subsequent research by I. Waldspurger in (Waldspurger 2017) relaxed the admissibility conditions on the wavelets used in the transform. This chapter draws upon these two works, as well as (J. Bruna 2022).

4.1 The Scattering Representation

Let $\psi \in \boldsymbol{L}^2(\mathbb{R}^d)$ be a mother wavelet with zero mean

$$\int_{\mathbb{R}^d} \psi(u)\, du = 0$$

and a fast decay. Figure 4.1 shows an example of such a mother wavelet for $d = 1$. The first step to built a representation for an input signal $\boldsymbol{x} \in \boldsymbol{L}^2(\mathbb{R}^d)$ is to convolute it with rotations and dilations of ψ.

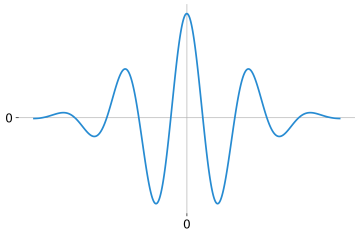


Figure 4.1: The Morlet wavelet is a Gaussian window modulated by a sinusoid. It is commonly used to define the scattering transform.

Let $a \in \mathbb{R}$ with $a > 1$ be a scale factor that generates a sequence $\{a^j\}_{j\in\mathbb{Z}}$. For the purpose of this case and to simplify notation, we set $a = 2$, which is a common choice for image processing. Let G be a group of rotations of $\mathbb{R}^d$ and let $\lambda = 2^j r \in \Lambda = 2^{\mathbb{Z}} \times G$ with $|\lambda| = 2^j$ be the parameter of the daughter wavelets. A dilation by 2^j and a rotation by $r \in G$ of the mother wavelet generate the daughter wavelet

$$\psi_\lambda(u) = |\lambda|^{-d}\psi(\lambda^{-1}u). \tag{4.1}$$

The filter bank $\{W_\lambda \boldsymbol{x}(u) = \boldsymbol{x} * \psi_\lambda(u)\}_{\lambda\in\hat{\Lambda}}$ gives a redundant representation of the signal. If $\boldsymbol{x}$ and ψ are real, $W_{-\lambda}\boldsymbol{x} = W_\lambda \boldsymbol{x}^* = W_\lambda \boldsymbol{x}$, and hence we can restrict us to the quotient group $G^+ = G/\{-1, 1\}$ of positive rotations. To ensure these conditions, we will be interested in mother wavelets of the form $\psi(u) = e^{i\eta\cdot u}\theta(u)$, where $\eta \in \mathbb{R}^d$ and θ is a lowpass

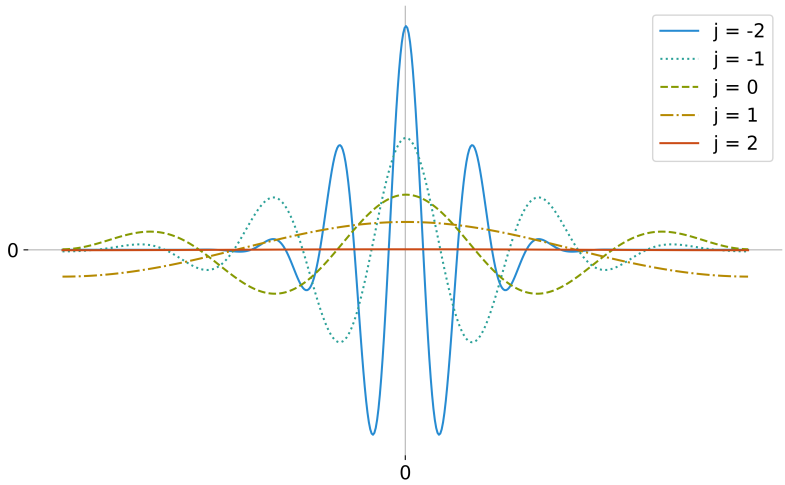


Figure 4.2: ψ_λ with diferent dilation factors when ψ is a real-valued Morlet wavelet.

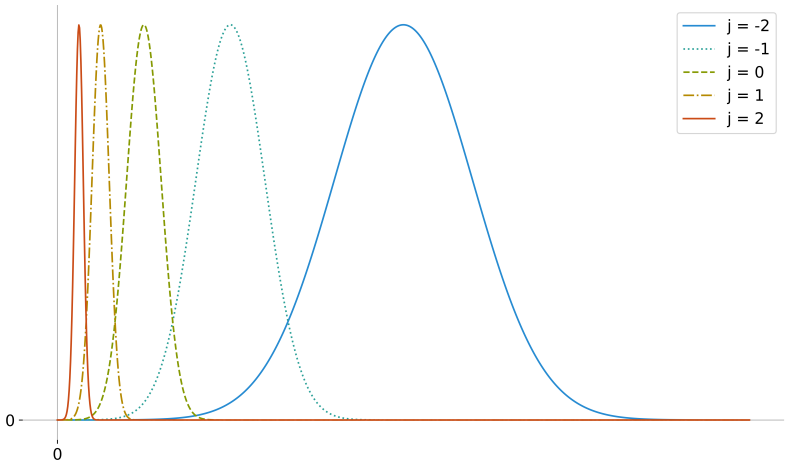


Figure 4.3: $\hat{\psi}_\lambda$ with diferent dilation factors when ψ is a real-valued Morlet wavelet.

window such that $\hat{\theta}$ is real. Thus, $\hat{\psi}(w) = \hat{\theta}(w - \eta)$ and $\hat{\psi}_\lambda(w) = \hat{\theta}(\lambda w - \eta)$ will be real.

Figure 4.2 and Figure 4.3 show daugther wavelets and their Fourier transform for different values of λ in the $d = 1$ case, where G^+ is the trivial group. The normalization factor in the definition of ψ_λ ensures that $\hat{\psi}_\lambda(w) = \hat{\psi}(\lambda w)$, as can be observed in Figure 4.3.

The wavelet transform of $\boldsymbol{x}$ is the redundant representation $\{\boldsymbol{x} * \psi_\lambda\}_{\lambda \in \Lambda}$. These wavelet coefficients are not translation invariant, because the convolutions translate as the input signal is translated. With this in mind, we use the complex modulus and compute $U[\lambda]\boldsymbol{x} = |\boldsymbol{x} * \psi_\lambda|$. This procedure is iterated by computing new wavelet decompositions across paths. A sequence $p = (\lambda_1, \ldots, \lambda_m) \in \Lambda^m$ defines a path, which is an ordered product of non-linear and

non-commuting operators

$$\begin{aligned} U[p]\boldsymbol{x} &= U[\lambda_m] \dots U[\lambda_2] U[\lambda_1]\boldsymbol{x} \\ &= |\dots||\boldsymbol{x} * \psi_{\lambda_1}| * \psi_{\lambda_2}| \cdots * \psi_{\lambda_m}|, \end{aligned}$$

with $U[\emptyset] = \mathrm{Id}$.

A wavelet transform with a finite scale 2^J only considers dilations up to that scale, this is ψ_λ with $|\lambda| \leq 2^J$. The remaining wavelets do not capture frequencies lower than 2^{-J}, so higher values of J increase the captured range of frequencies. To recover the low frequencies, we use a low-pass filter ϕ_J with a spatial support proportional to 2^J. To define this filter, we consider a real-valued and positive function ϕ and define the scaled spatial window

$$\phi_J(u) = 2^{-dJ}\phi(2^{-J}u). \tag{4.2}$$

Then, we define the scattering transform of order m as

$$S_J[\mathcal{P}_J]\boldsymbol{x} = \{S_J[p]\boldsymbol{x} = U[p]\boldsymbol{x} * \phi_J\}_{p \in \mathcal{P}_J} \tag{4.3}$$

where $\mathcal{P}_J$ is the set of paths of length less or equal than m and elements in Λ_J^+. This procedure restores averages that are translation-invariant and stable to the action of diffeomorphisms. For each path p, the coefficient $S_J[p]\boldsymbol{x}(u)$ can be subsampled at intervals proportional to the window size 2^J.

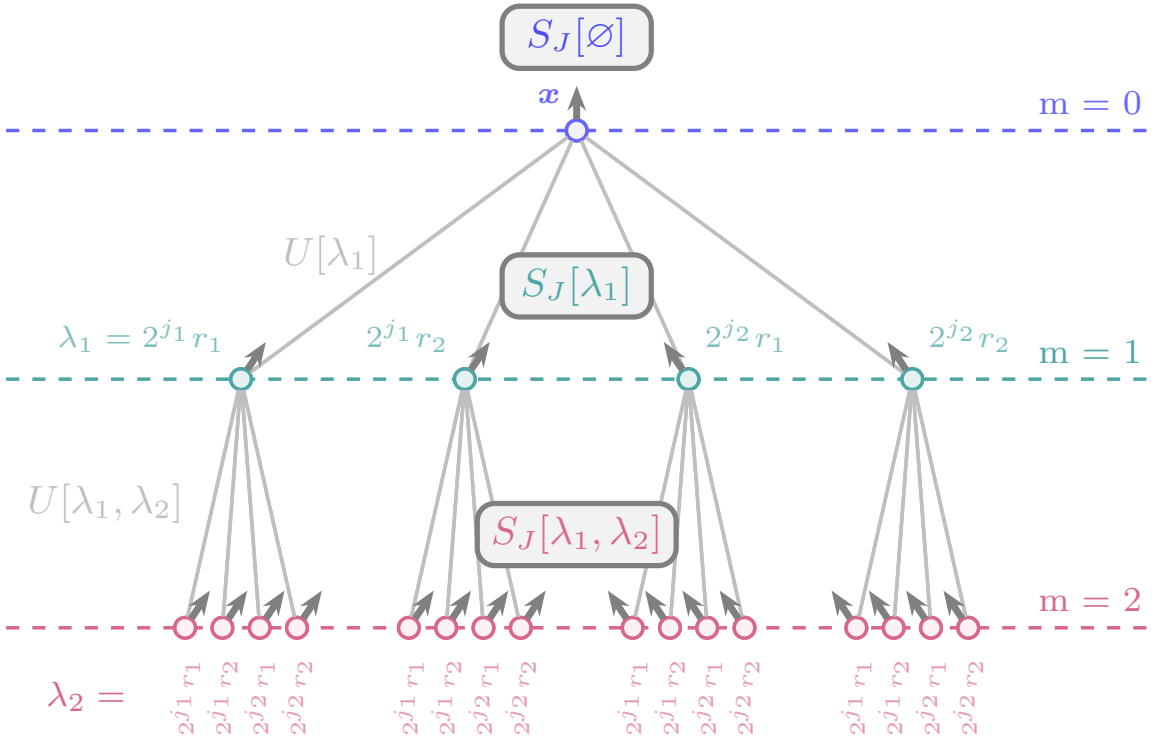


Figure 4.4: Illustration of the scattering transform for the coefficients $S_J[p]\boldsymbol{x}$ up to order $m = 2$ and $\Lambda = (j_1, j_2) \times (r_1, r_2)$.

4.2 Properties of the Wavelet Transform

In this section, we are going to see that the wavelet transform is both stable and complete. Consider the wavelet transform

$$\mathcal{W}_J \boldsymbol{x} = \{\boldsymbol{x} * \phi_J, \{W_\lambda \boldsymbol{x}\}_{\lambda \in \Lambda_J}\} \tag{4.4}$$

with norm defined as

$$\|\mathcal{W}_J \boldsymbol{x}\|_2^2 = \|\boldsymbol{x} * \phi_J\|_2^2 + \sum_{\lambda \in \Lambda_J} \|W_\lambda \boldsymbol{x}\|_2^2. \tag{4.5}$$

Then,

$$\mathcal{W}_J : \boldsymbol{L}^2(\mathbb{R}^d) \to \boldsymbol{L}^2(\mathbb{R}^d) \times \prod_{\lambda \in \Lambda_J} \boldsymbol{L}^2(\mathbb{R}^d) \tag{4.6}$$

is a linear operator to a product space of copies of $\boldsymbol{L}^2(\mathbb{R}^d)$. The linearity of $\mathcal{W}_J$ comes from the linearity of the convolution operator. Now, we are going to see that under certain conditions, this operator defines a frame of $\boldsymbol{L}^2(\mathbb{R}^d)$.

Definition 4.2.1 *Let V be an inner product space. The set of vectors $\{v_i\}_{i\in\mathcal{I}} \subset V$ with $\mathcal{I} \subset \mathbb{N}$ is a frame for V if there exist $\alpha, \beta \in \mathbb{R}$ such that $0 < \alpha \leq \beta < +\infty$ and*

$$\alpha\|u\|^2 \leq \sum_{i\in\mathcal{I}} |\langle u, v_i\rangle|^2 \leq \beta\|u\|^2 \tag{4.7}$$

for every $u \in V$.

We need that the frequency supports of the wavelets cover uniformly the whole frequency domain. With this, we can show that $\mathcal{W}_J\boldsymbol{x}$ defines a frame.

Proposition 4.2.1 *If there exists $0 \leq \epsilon < 1$ such that*

$$1 - \epsilon \leq |\hat{\phi}_J(\omega)|^2 + \sum_{\lambda\in\Lambda_J} |\hat{\psi}_\lambda(\omega)|^2 \leq 1 \tag{4.8}$$

for almost every $\omega \in \mathbb{R}^d$, then $\mathcal{W}_J$ is a frame of $\boldsymbol{L}^2(\mathbb{R}^d)$ with

$$(1 - \epsilon)\|\boldsymbol{x}\|_2^2 \leq \|\mathcal{W}_J\boldsymbol{x}\|_2^2 \leq \|\boldsymbol{x}\|_2^2 \tag{4.9}$$

for every $\boldsymbol{x} \in \boldsymbol{L}^2(\mathbb{R}^d)$.

Proof. Let $w_0 = \phi_J$ and $w_{\iota(\lambda)} = \psi_\lambda$ for some injection $\iota : \Lambda_J \hookrightarrow \mathbb{N} \setminus 0$. Let $\mathcal{I}_J$ denote the set of indices $\iota(\Lambda_J) \cup 0$. The set of indices $\mathcal{I}_J$ might be finite or infinite. In any case, we are concerned with the space of finite energy coefficients $\boldsymbol{\ell}^2(\mathcal{I}_J) = \{u : \sum_{i\in\mathcal{I}_J} \|u_i\|_2^2 < +\infty\}$. Then, we can use the notation $\mathcal{W}_J\boldsymbol{x} = (\boldsymbol{x} * w_i)_{i\in\mathcal{I}_J}$ for the wavelet transform and for its norm

$$\|\mathcal{W}_J\boldsymbol{x}\|_2^2 = \sum_{i\in\mathcal{I}_J} \|\boldsymbol{x} * w_i\|_2^2.$$

With the introduced notation, we can rewrite Equation 4.8 and Equation 4.9 as

$$1 - \epsilon \leq \sum_{i\in\mathcal{I}_J} |\hat{w}_i(\omega)|^2 \leq 1 \tag{4.10}$$

and

$$(1 - \epsilon)\|\boldsymbol{x}\|_2^2 \leq \sum_{i\in\mathcal{I}_J} \|\boldsymbol{x} * w_i\|_2^2 \leq \|\boldsymbol{x}\|_2^2. \tag{4.11}$$

Using Parseval's theorem

$$\|\boldsymbol{x} * w_i\|_2^2 = \|\hat{\boldsymbol{x}} \cdot \hat{w}_i\|_2^2 = \int_{\mathbb{R}^d} |\hat{\boldsymbol{x}}(\omega)|^2 |\hat{w}_i(\omega)|^2 \, d\omega$$

Then

$$\begin{aligned}\sum_{i \in \mathcal{I}_J} \|\boldsymbol{x} * w_i\|_2^2 &= \sum_{i \in \mathcal{I}_J} \int_{\mathbb{R}^d} |\hat{\boldsymbol{x}}(\omega)|^2 |\hat{w}_i(\omega)|^2 \, d\omega \\ &= \int_{\mathbb{R}^d} |\hat{\boldsymbol{x}}(\omega)|^2 \sum_{i \in \mathcal{I}_J} |\hat{w}_i(\omega)|^2 \, d\omega\end{aligned}$$

and using Equation 4.10 we obtain Equation 4.11. We call this a frame even if this result is not exactly the same as the definition of frame stated in Equation 4.7, but because it frames the norm of the operator $\mathcal{W}_J \boldsymbol{x}$. □

The condition in Proposition 4.2.1 is a so-called Littlewood-Paley condition and this result proves that the wavelet transform is complete and stable. The completeness comes from Equation 4.8 because this, together with $\hat{\psi}_\lambda(w) = \hat{\psi}(\lambda w)$, implies that the whole frequency domain is covered by the dilations and rotations of $\hat{\psi}$. Hence, the signal $\boldsymbol{x}$ can be recovered from its wavelet transform. The stability comes from Equation 4.9. Note that the wavelet transform is redundant, so that a sequence of coefficients is not necessarily the wavelet transform of some signal.

4.3 Properties of the Scattering Representation

In this section, we review and prove several key properties of the scattering transform.

4.3.1 Stability to Additive Noise

By leveraging the properties of the wavelet transform discussed earlier, we can establish the stability of the scattering transform to additive noise. In addition to the references mentioned earlier, we also consulted (S. Mallat and Zhong 1992) when preparing this section.

Let us start by seeing that $\mathcal{W}_J$ is a non-expansive operator.

Proposition 4.3.1 *If, for every* $\boldsymbol{x} \in \boldsymbol{L}^2(\mathbb{R}^d)$,

$$(1-\epsilon)\|\boldsymbol{x}\|_2^2 \leq \|\mathcal{W}_J\boldsymbol{x}\|_2^2 \leq \|\boldsymbol{x}\|_2^2, \tag{4.12}$$

then $\mathcal{W}_J$ *is a non-expansive operator.*

Proof. $\|\mathcal{W}_J\boldsymbol{x}-\mathcal{W}_J\boldsymbol{x'}\|_2^2 = \|\mathcal{W}_J(\boldsymbol{x}-\boldsymbol{x'})\|_2^2 \leq \|\boldsymbol{x}-\boldsymbol{x'}\|_2^2$. Here we have used the linearity of the operator and the frame condition proved before. □

Now, define the propagator operator

$$\mathcal{U}_J\boldsymbol{x} = \{\boldsymbol{x} * \phi_J, \{U[\lambda]\boldsymbol{x}\}_{\lambda\in\Lambda_J}\}, \tag{4.13}$$

such that the first layer of the scattering representation is obtained by applying $\mathcal{U}_J$ to the input signal and the successive layers are obtained by applying $\mathcal{U}_J$ to the output of the previous layer, so that

$$\begin{aligned}\mathcal{U}_J U[p]\boldsymbol{x} &= \{U[p]\boldsymbol{x} * \phi_J, \{U[p+\lambda]\boldsymbol{x}\}_{\lambda\in\Lambda_J}\} &(4.14)\\ &= \{S_J[p]\boldsymbol{x}, \{U[p+\lambda]\boldsymbol{x}\}_{\lambda\in\Lambda_J}\}, &(4.15)\end{aligned}$$

where we have used that $U[\lambda]U[p] = U[p+\lambda]$.

We can see now that this operator is also non-expansive.

Proposition 4.3.2 $\mathcal{U}_J$ *is a non-expansive operator.*

Proof. $\|\mathcal{U}_J\boldsymbol{x}-\mathcal{U}_J\boldsymbol{x'}\|_2^2 = \|\boldsymbol{x}*\phi_J-\boldsymbol{x'}*\phi_J\|_2^2+\sum_{\lambda\in\Lambda_J}\||\boldsymbol{x}*\psi_\lambda|-|\boldsymbol{x'}*\psi_\lambda|\|_2^2 \leq \|\boldsymbol{x}*\phi_J-\boldsymbol{x'}*\phi_J\|_2^2+\sum_{\lambda\in\Lambda_J}\|\boldsymbol{x}*\psi_\lambda-\boldsymbol{x'}*\psi_\lambda\|_2^2 = \|\mathcal{W}_J\boldsymbol{x}-\mathcal{W}_J\boldsymbol{x'}\|_2^2 \leq \|\boldsymbol{x}-\boldsymbol{x'}\|_2^2$, where we have used that the modulus is non-expansive, so that $||u|-|v|| \leq |u-v|$, and the result of Proposition 4.3.1. □

With this, we can prove that the scattering transform is also non-expansive and stable to additive noise, defining its norm as

$$\|S_J[P]\boldsymbol{x}\|_2^2 = \sum_{p\in P}\|S_J[p]\boldsymbol{x}\|_2^2, \tag{4.16}$$

where P is a set of paths.

Proposition 4.3.3 *The scattering transform is a non-expansive operator.*

Proof. $S_J[\mathcal{P}_J]$ is constructed by propagating $\mathcal{U}_J$ as shown in Equation 4.15. Since $\mathcal{U}_J$ is a non-expansive operator, $S_J[\mathcal{P}_J]$ is also non-expansive. □

Proposition 4.3.4 *The scattering transform is stable to additive noise.*

Proof. By the fact that the scattering transform is non-expansive, $\|S_J[\mathcal{P}_J]\boldsymbol{x} - S_J[\mathcal{P}_J](\boldsymbol{x}+\boldsymbol{\epsilon})\|_2^2 \leq \|\boldsymbol{\epsilon}\|_2^2$. □

4.3.2 Energy Conservation

A wavelet decomposition is called unitary if the wavelets satisfy the condition in Proposition 4.2.1 with $\epsilon = 0$. In that case,

$$\|\mathcal{W}_J\boldsymbol{x}\|_2^2 = \|\boldsymbol{x}\|_2^2. \tag{4.17}$$

Proposition 4.3.5 *If the wavelet decomposition is unitary, then*

$$\|\boldsymbol{x}\|_2^2 = \sum_{|p|<m} \|S_J[p]\boldsymbol{x}\|_2^2 + \sum_{|p|=m} \|U[p]\boldsymbol{x}\|_2^2 \tag{4.18}$$

for any order $m \in \mathbb{N}$.

Proof. We will prove it by induction on the order m. The base case is $m = 0$, when

$$\sum_{|p|<0} \|S_J[p]\boldsymbol{x}\|_2^2 + \sum_{|p|=0} \|U[p]\boldsymbol{x}\|_2^2 = \|U[\emptyset]\boldsymbol{x}\|_2^2 = \|\boldsymbol{x}\|_2^2$$

and hence Equation 4.18 is true for $m = 0$. For the induction step, assume it is true for $m = k \in \mathbb{N}$. Then

$$\begin{aligned}&\sum_{|p|<k+1} \|S_J[p]\boldsymbol{x}\|_2^2 + \sum_{|p|=k+1} \|U[p]\boldsymbol{x}\|_2^2 \\ &= \sum_{|p|<k} \|S_J[p]\boldsymbol{x}\|_2^2 + \sum_{|p|=k} \|S_J[p]\boldsymbol{x}\|_2^2 + \sum_{|p|=k+1} \|U[p]\boldsymbol{x}\|_2^2.\end{aligned}$$

Observe that

$$\begin{aligned}
&\sum_{|p|=k} \|S_J[p]\boldsymbol{x}\|_2^2 + \sum_{|p|=k+1} \|U[p]\boldsymbol{x}\|_2^2 \\
&= \sum_{|p|=k} [\|U[p]\boldsymbol{x} * \phi_J\|_2^2 + \sum_{\lambda\in\Lambda_J} \|U[\lambda]U[p]\boldsymbol{x}\|_2^2] \\
&= \sum_{|p|=k} \|\mathcal{U}_J U[p]\boldsymbol{x}\|_2^2 = \sum_{|p|=k} \|\mathcal{W}_J U[p]\boldsymbol{x}\|_2^2 \\
&= \sum_{|p|=k} \|U[p]\boldsymbol{x}\|_2^2,
\end{aligned}$$

where we have used that the wavelet decomposition is unitary and

$$\begin{aligned}
\|\mathcal{U}_J\boldsymbol{x}\|_2^2 &= \|\boldsymbol{x} * \phi_J\|_2^2 + \sum_{\lambda\in\Lambda_J} \|U_\lambda\boldsymbol{x}\|_2^2 \\
&= \|\boldsymbol{x} * \phi_J\|_2^2 + \sum_{\lambda\in\Lambda_J} \||W_\lambda\boldsymbol{x}|\|_2^2 \\
&= \|\boldsymbol{x} * \phi_J\|_2^2 + \sum_{\lambda\in\Lambda_J} \|W_\lambda\boldsymbol{x}\|_2^2 \\
&= \|\mathcal{W}_J\boldsymbol{x}\|_2^2.
\end{aligned}$$

Thus, Equation 4.18 holds for $m = k + 1$. By the principle of induction, it holds for all $m \in \mathbb{N}$. □

The previous proposition decomposes the energy of the signal into two terms: the energy captured by the first m layers of the scattering transform, and a residual energy term that we define as follows.

Definition 4.3.1 *Given a signal $\boldsymbol{x} \in \boldsymbol{L}^2(\mathbb{R}^d)$, the residual energy of its scattering transform of order m is*

$$\mathcal{R}_J(m) = \sum_{|p|=m} \|U[p]\boldsymbol{x}\|_2^2.$$

Our objective at this point is to characterize the residual energy, which can serve as a guide when determining the appropriate number of layers needed to capture the bulk of the signal's energy. For this, we first define a Gaussian function as follows.

Definition 4.3.2 *For any $\alpha > 0$, χ_α is the Gaussian function*

$$\chi_\alpha(t) = \sqrt{\pi}\alpha \exp(-(\pi\alpha t)^2)$$

for $t \in \mathbb{R}$, with Fourier transform

$$\hat{\chi_\alpha}(\omega) = \exp(-\frac{\omega^2}{\alpha^2})$$

for $\omega \in \mathbb{R}$.

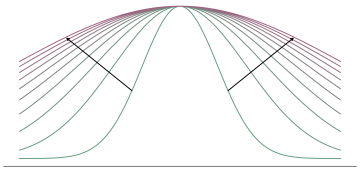


Figure 4.5: Fourier transform $\hat{\chi_\alpha}$ of the Gaussian function χ_α. Increasing the value of the parameter α results in a wider Fourier transform, as indicated by the arrows in the diagram.

The one-dimensional scattering transform's residual energy is bounded by the following theorem by (Waldspurger 2017).

Theorem 4.3.6 *For the $d = 1$ case, let $\{\psi_\lambda\}_{\lambda \in \Lambda_J}$ be a family of wavelets. First, assume that*

$$\sum_{\lambda \in \Lambda_J} \left(|\hat{\psi_\lambda}(\omega)|^2 + |\hat{\psi_\lambda}(-\omega)|^2 \right) \leq 2 \tag{4.19}$$

for every $\omega \in \mathbb{R}$. Second, assume that

$$|\hat{\psi_\lambda}(-\omega)| \leq |\hat{\psi_\lambda}(\omega)| \tag{4.20}$$

for every $\lambda \in \Lambda_J$ and $\omega > 0$ and that for any such ω, exists $\lambda \in \Lambda_J$ such that the inequality is strict. Third, assume that there exists $\epsilon > 0$ such that

$$\hat{\psi}(\omega) = O(|\omega|^{1+\epsilon}) \tag{4.21}$$

when $\omega \to 0$. Then, for any $J \in \mathbb{Z}$, there exist $r > 0$ and $a > 1$ such that

$$\mathcal{R}_J(m) \leq \|\boldsymbol{x}\|_2^2 - \|\boldsymbol{x} * \chi_{ra^m}\|_2^2 \tag{4.22}$$

for every $m \geq 2$ and $\boldsymbol{x} \in \boldsymbol{L}^2(\mathbb{R})$.

Equation 4.19 is a Littlewood-Paley condition that ensures the wavelet transform is non-expansive, preventing it from amplifying certain frequencies of $\boldsymbol{x}$, and causing the energy in m-length paths to decrease as m increases.

Equation 4.20 specifies that the wavelets must give greater weight to positive frequencies than to the negative ones, which guarantees that the energy of the signal shifts towards the low frequencies.

Equation 4.21 guarantees that the wavelets have more than one zero momentum. This is a prerequisite for the proof in (Waldspurger 2017), but may not be necessary.

For Equation 4.22, using Parseval's theorem, we observe that

$$\|\boldsymbol{x} * \chi_{ra^m}\|_2^2 = \|\hat{\boldsymbol{x}} \cdot \hat{\chi}_{ra^m}\|_2^2.$$

Hence, the Fourier transform $\hat{\chi}_{ra^m}$ covers a broader frequency range as m increases, and the residual energy bound approaches zero.

This theorem establishes the energy conservation of the scattering transform and proves an exponential rate of

energy decay through the scattering tree. In particular, the energy in $\boldsymbol{x}$ coming from frequencies around 2^m is captured after $O(m)$ layers. We do not include its proof as it is extense and highly technical.

It is believed that this theorem can be extended to $\boldsymbol{L}^2(\mathbb{R})^d$ for $d \geq 1$. A previous theorem in (S. Mallat 2012) guarantees energy conservation for any $d \geq 1$, but it imposes stricter conditions on the wavelets and does not provide an explicit convergence rate. These results explain why in applications we can discard the high-order scattering coefficients. In our case, we will consider coefficients up to order $m = 2$.

4.3.3 Translation Invariance

To prove the translation invariant property of the scattering transform, we first define an admissibility condition on the family of wavelets used.

Definition 4.3.3 *A mother wavelet ψ is admissible if there exist $\eta \in \mathbb{R}^d$ and $\rho \geq 0$, with $\hat{\rho}(0) = 1$ and $|\hat{\rho}(\omega)| \leq |\hat{\phi}(2\omega)|$, such that the function Ψ defined as*

$$\hat{\Psi}(\omega) = |\hat{\rho}(\omega - \eta)|^2 - \sum_{j=1}^{+\infty} j \left(1 - |\hat{\rho}(2^{-j}\omega - 2^{-j}\eta)|^2\right)$$

satisfies

$$\inf_{1 \leq |\omega| \leq 2} \sum_{j=-\infty}^{+\infty} \sum_{r \in G} \hat{\Psi}(2^{-j} r^{-1} \omega) |\hat{\psi}(2^{-j} r^{-1} \omega)|^2 > 0.$$

Let $\boldsymbol{x}_\tau(u) = \boldsymbol{x}(u - \tau)$ denote the translation of $\boldsymbol{x} \in \boldsymbol{L}^2(\mathbb{R}^d)$ by $\tau \in \mathbb{R}^d$. In (S. Mallat 2012), the following proposition is proved.

Proposition 4.3.7 *For all $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{L}^2(\mathbb{R}^d)$ and $J \in \mathbb{Z}$,*

$$\|S_{J+1}[\mathcal{P}_{J+1}]\boldsymbol{x} - S_{J+1}[\mathcal{P}_{J+1}]\boldsymbol{y}\|_2^2 \leq \|S_J[\mathcal{P}_J]\boldsymbol{x} - S_J[\mathcal{P}_J]\boldsymbol{y}\|_2^2.$$

This implies that as J increases, the distance between two signals in the scattering domain does not increase. As it is positive and non-increasing, as J approaches infinity it converges and defines a limit distance that is translation

invariant. The last result is formalized in (S. Mallat 2012) as the following theorem.

Theorem 4.3.8 *Let $\{\psi_\lambda\}_\lambda$ be a family of wavelets with a mother wavelet ψ that satisfies the admissibility condition. Then, for any $\boldsymbol{x} \in \mathbf{L}^2(\mathbb{R}^d)$ and $\tau \in \mathbb{R}^d$,*

$$\lim_{J \to +\infty} \|S_J[\mathcal{P}]\boldsymbol{x} - S_J[\mathcal{P}]\boldsymbol{x}_\tau\|_2^2 = 0$$

The translation invariance lies in the fact that wavelet modulus decompositions translate along with the input signal, and the presence of the averaging operator in the scattering transform.

4.3.4 Lipschitz Continuity under the Action of Diffeomorphisms

Lastly, we look at the Lipschitz continuity of the scattering transform under the action of $\mathcal{C}^2$ diffeomorphisms. A diffeomorphism τ acts on a signal $\boldsymbol{x} \in \boldsymbol{L}^2(\mathbb{R}^d)$ mapping it to $\boldsymbol{x}_\tau(u) = \boldsymbol{x}(u - \tau(u))$. (S. Mallat 2012) states the following theorem.

Theorem 4.3.9 *Let $\Omega \subset \mathbb{R}^d$ be a compact set. Then, there exists C such that for all $\boldsymbol{x} \in \boldsymbol{L}^2(\mathbb{R}^d)$ (compactly) supported in Ω with $\|U[\mathcal{P}]\boldsymbol{x}\|_1 < \infty$ and for all $\tau \in \mathcal{C}^2(\mathbb{R}^d)$ with $\|\nabla\tau\|_\infty \leq 1/2$, if $\|\tau\|_\infty \leq 2^J\|\nabla\tau\|_\infty$, then*

$$\|S_J[\mathcal{P}]\boldsymbol{x}_\tau - S_J[\mathcal{P}]\boldsymbol{x}\|_2 \leq C\|U[\mathcal{P}]\boldsymbol{x}\|_1 \left(\|\nabla\tau\|_\infty + \|H\tau\|_\infty\right),$$

where H denotes the Hessian matrix.

The proof can be found in (S. Mallat 2012), where an upper bound for $\|S_J[\mathcal{P}]\boldsymbol{x}_\tau - S_J[\mathcal{P}]\boldsymbol{x}\|_2$ in a more general case is also provided.

Implementation of the Scattering Transform

5

We are interested in computing the scattering transform of images with a single channel. This can be achieved using Kymatio, a Python implementation of the scattering transform. Kymatio is highly efficient and it can be used for large-scale experiments in machine learning. It is compatible with PyTorch, TensorFlow, JAX, and other APIs, and can be run on both CPUs and GPUs. Additionally, Kymatio supports 1, 2, and 3-dimensional wavelet filter banks, as described in (Andreux, Angles et al. 2020).

For this project, we have developed a Python implementation of the scattering transform for learning purposes, using 2-dimensional wavelets and specifically for image analysis. We built this implementation with PyTorch and based its main algorithms on Kymatio. All the code and documentation for our implementation can be accessed in the scattering repository, which also offers tools for image synthesis, super-resolution, and other functionalities relevant to our modeling task. We plan to expand the implementation in the future to incorporate a phase harmonic covariances representation.

5.1 Scattering Representation of Images

This section elaborates on the implementation of the scattering transform for the case where $d = 2$, which was previously introduced in Chapter 4. Table 5.1 shows the possible values for the parameters of the scattering representation in our implementation.

Table 5.1: Symbol and domain of the parameters of the scattering representation.

Parameter	Symbol	Domain
Scale	J	$1 \leq J \leq \log_2(\min(N, M))$
Number of rotations	L	$L \geq 1$
Maximum order	m^*	$0 \leq m^* \leq 2$

Let us consider signals defined on a grid of dimensions $N \times M$. We can represent the periodic convolution of two signals using the notation $x \circledast y[n, m]$. Additionally, let

$\phi_J[n,m]$ denote a lowpass filter with an averaging scale of 2^J. We introduce a wavelet filter bank $\psi_\lambda[n,m]_{\lambda\in\Lambda}$ with $\Lambda = j : 1 \leq j \leq J \times G^+$, where G^+ represents a set of L rotations uniformly distributed in a semicircle. Then, the 0-order scattering coefficient is

$$S_0 x[n,m] = x \circledast \phi_J[n,m]. \tag{5.1}$$

The 1-order scattering coefficients are

$$S_1[\lambda]x[n,m] = |x \circledast \psi_\lambda| \circledast \phi_J[n,m] \qquad \lambda \in \Lambda. \tag{5.2}$$

The 2-order scattering coefficients are

$$S_2[\lambda_1,\lambda_2]x[n,m] = ||x \circledast \psi_{\lambda_1}| \circledast \psi_{\lambda_2}| \circledast \phi_J[n,m] \qquad \lambda_1,\lambda_2 \in \Lambda \tag{5.3}$$

To minimize the size of the representation, we select wavelets with $|\lambda_2| > |\lambda_1|$ since $|x \circledast \psi_{\lambda_1}|$ is a low-frequency signal. This is because most of the signal's energy is concentrated in the paths with strictly decreasing frequency. Additionally, as per Theorem 4.3.6, high-order coefficients generally have negligible energy. Hence, we follow Kymatio's approach and compute coefficients up to second order only. To further reduce memory requirements, we downsample the filtered low-frequency signals.

Hence, the computed scattering transform with a scale factor J and L rotations of a $N \times M$ image will be a tensor with shape

$$\left[1 + LJ + L^2\binom{J}{2}\right] \times \frac{N}{2^J} \times \frac{M}{2^J} \tag{5.4}$$

The first term corresponds to the number of paths, and the second and third terms come from to the downsampling. Figure 5.1 shows the total size of the scattering representation as we increase J. Although the number of coefficients $1 + LJ + L^2\binom{J}{2}$ increases with J, the size of every coefficient $\frac{N}{2^J} \times \frac{M}{2^J}$ decreases exponentially. Table 5.2 displays the total size of the scattering transform for low-resolution and high-resolution images and various combinations of parameters.

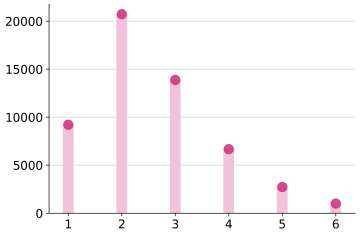


Figure 5.1: Size of the scattering representation for different values of J, with $m = 2$, $L = 8$ and $N = M = 64$.

There is only one coefficient of order 0. For 1-order coefficients, there are LJ options for the parameter λ. When it comes to 2-order coefficients, we have to choose (j_1, r_1) and (j_2, r_1) such that $j_1 > j_2$. Therefore, there are $L^2\binom{J}{2}$ coefficients of order 2.

J	L = 4	L = 8
1	5120	9216
2	6400	20736
3	3904	13888
4	1808	6672
5	724	2724
6	265	1009

J	L = 4	L = 8
1	81920	147456
2	102400	331776
3	62464	222208
4	28928	106752
5	11584	43584
6	4240	16144
7	1460	5604
8	481	1857

Table 5.2: Total size of the scattering representation of an image for various combinations of J and L. The left table displays the results for 64×64 images, while the right table shows them for 256×256 images. Any representations that exceed the corresponding image size of $64^2 = 4096$ or $256^2 = 65536$ are highlighted in magenta.

Filters

Morlet filters are used to analyze whether there is any information of a specific frequency in an image within a particular direction. These filters are built by modulating a Gaussian function with a sinusoidal wave. We use Morlet wavelets of the form

$$g(x, y; \sigma, \theta, \xi, \gamma) = \frac{\gamma}{2\pi\sigma^2} \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\xi x'\right), \tag{5.5}$$

where $x' = x\cos\theta + y\sin\theta$ and $y' = y\cos\theta - x\sin\theta$. In this expression, σ corresponds to the standard deviation of the Gaussian envelope, θ specifies the orientation of the parallel stripes, ξ is the wavenumber of the sinusoidal wave and γ determines the ellipticity of the filter.

The wavelets ψ_λ are defined as in Kymatio using complex Morlet wavelets. Figure 5.2 and Figure 5.3 illustrate how these filters look like for different rotations and dilation factors choices. Figure 5.5 shows the Fourier transforms of these wavelets, which are real signals as explained in Chapter 4. The dilations and rotations of the mother wavelet cover a semicircle in the frequency domain.

For the lowpass filters ϕ_J, we use real Gaussian envelopes without any modulation, resulting in positive signals. Figure 5.6 shows how the filters and their Fourier transforms look like for different values of J. As the filters are even real functions, their Fourier transforms are also real.

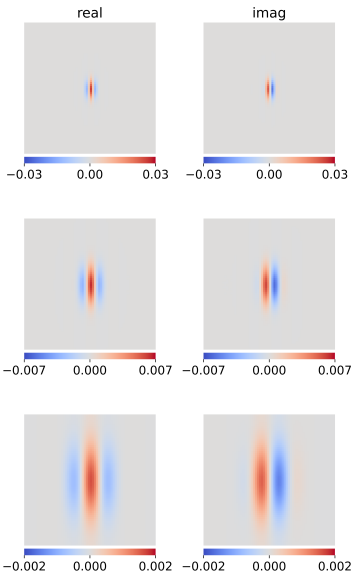
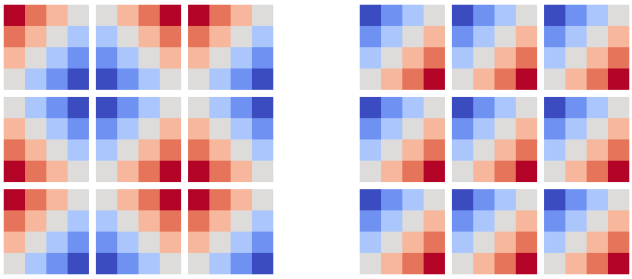


Figure 5.2: Real and imaginary parts of a Morlet wavelet filter. The scale factor j increases from the top to the bottom.

Padding

In order to mitigate boundary effects generated by the periodic convolutions computed with fast Fourier transforms,



(a) Reflection padding **(b)** Periodic padding

Figure 5.4: Illustration of the two implemented padding modes.

one can apply a padding to the input image. Figure 5.4 illustrates the two implemented padding modes. Although not applying padding would yield the same results as applying periodic padding due to the convolutions' periodic nature, the resulting extension would contain discontinuities that could impact the values at the boundaries of the coefficients.Thus, Kymatio utilizes reflection padding, which generates a more continuous extension that can be beneficial in many scenarios.

In this work, we focus on potential vorticity fields that have periodic boundary conditions, resulting in a highly continuous periodic extension. Although no padding maintains the signal size and is computationally advantageous, we must consider the potential impact of the padding choice. We implemented reflection padding solely for comparison purposes with Kymatio, which uses a reflection padding by default. Therefore, we did not prioritize the efficiency of this padding option. Applying no padding is, of course, a very efficient choice.

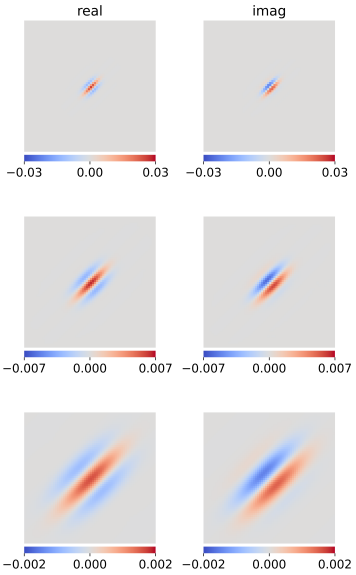


Figure 5.3: Real and imaginary parts of Morlet wavelet filter produced by rotating the filter from Figure 5.2. The scale factor j increases from the top to the bottom.

5.2 Analysis of the Implementation

To verify the correctness of our implementation of the scattering transform, we compared it with Kymatio, which has been thoroughly tested. Additionally, we analyzed the performance of our implementation on both a CPU and a GPU. CPUs are optimized for general-purpose computing tasks, while GPUs are designed to perform highly parallel computations on large datasets simultaneously. Vectorization involves performing operations on arrays or matrices instead of individual elements, and our implementation is

highly vectorized, which makes it particularly well-suited for GPUs.

5.2.1 Correctness

We evaluate the accuracy of our solution by comparing its outputs with those of Kymatio. Our evaluation set comprises 2000 low-resolution and 2000 high-resolution images, all featuring PyQG potential vorticity fields. Half of these images are from the top layer, and the other half are from the bottom layer. We calculate all coefficients up to the second order and compare them across a spectrum of values for both J and L.

Following (Oyallon, Zagoruyko et al. 2019), we quantify the relative error in the scattering domain as

$$re(S_J x) = \frac{\|S_J^k x - S_J x\|_2}{\|S_J^k x\|_2}, \tag{5.6}$$

where $S_J x$ represents our computation of the scattering coefficients for an image x and $S_J^k x$ represents the equivalent computation by Kymatio, which we use as the reference value. The relative error for the low-resolution images, across the entire range of possible J values and for three distinct L values, is depicted in Figure 5.7. The corresponding results for high-resolution images are shown in Figure 5.8. The relative error is plotted on a logarithmic scale.

Our approach yields coefficients that closely match those generated by Kymatio. Specifically, the norm of the difference falls between 3% and 0.001% of the norm of the reference coefficients. We attribute these discrepancies to minor variances in the filter definitions and numerical approximations within our method.

5.2.2 Performance on a CPU

Figure 5.9a and Figure 5.9b show the time comparison between Kymatio and our method for calculating the scattering coefficients on a CPU of low-resolution and high-resolution images, respectively. The observed difference is a result of our highly vectorized implementation, which significantly speeds up filter initialization, and the fact that

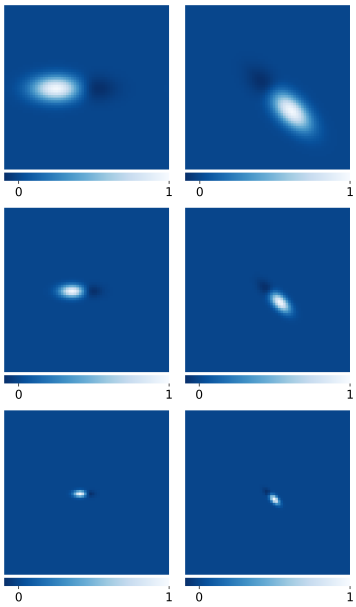


Figure 5.5: Fourier modulus of the Morlet wavelet filters from Figure 5.2 and Figure 5.3. The scale factor j increases from the top to the bottom.

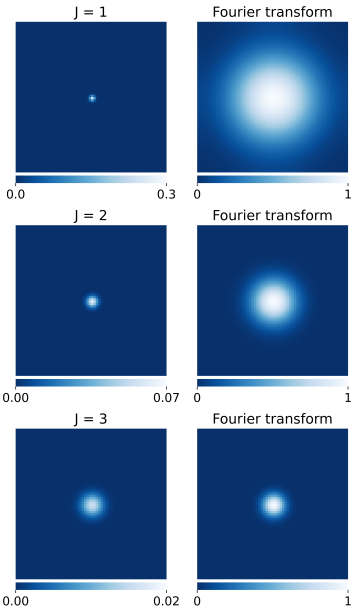


Figure 5.6: Lowpass filters ϕ and their Fourier transforms $\hat{\phi}$ for diferent scale factors J. All the signals are real.

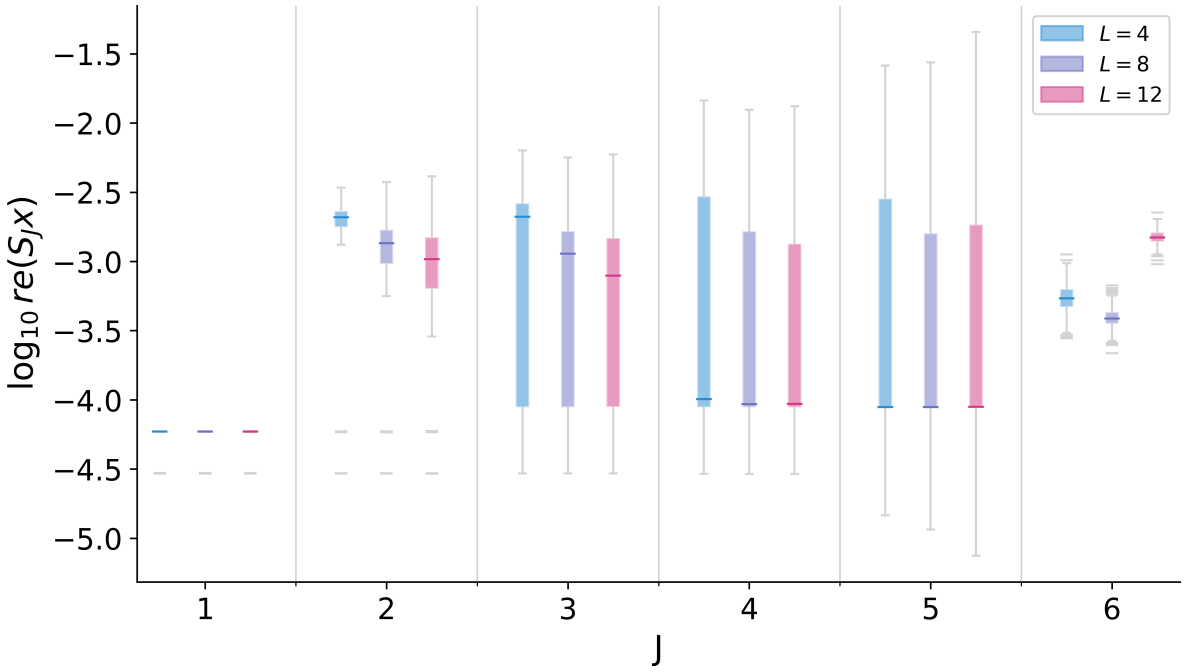


Figure 5.7: Relative error in the scattering domain (on a $\log_{10}$ scale) between our method's coefficients and Kymatio's coefficients for 2000 low-resolution images, plotted against the values of J and L.

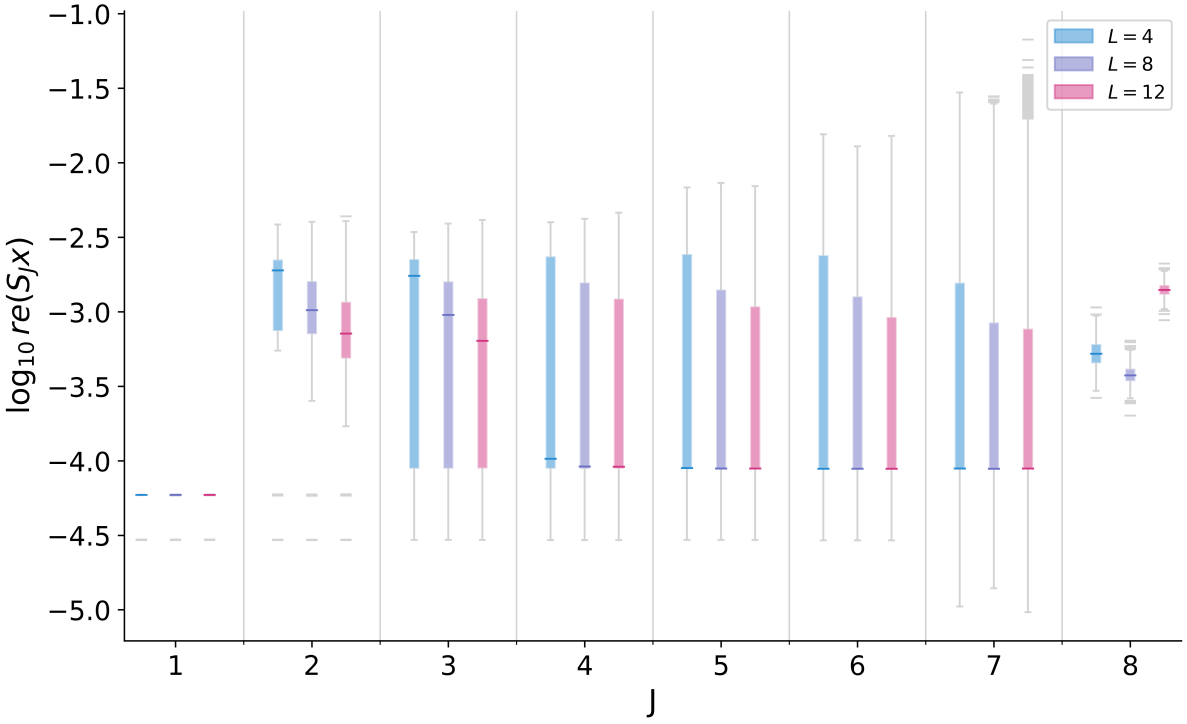
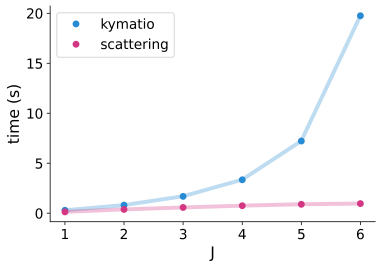
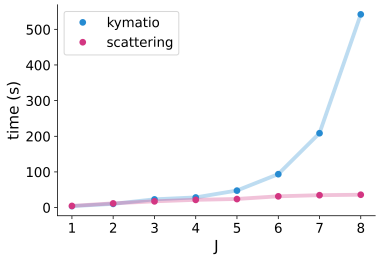


Figure 5.8: Relative error in the scattering domain (on a $\log_{10}$ scale) between our method's coefficients and Kymatio's coefficients for 2000 high-resolution images, plotted against the values of J and L.

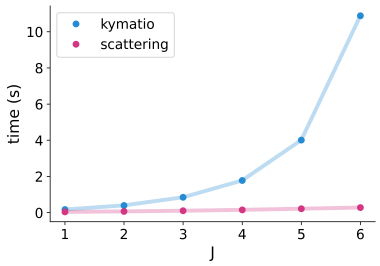


(a) Low-resolution images

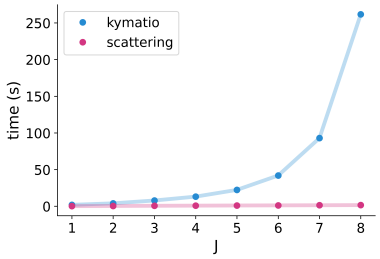


(b) High-resolution images

Figure 5.9: Time taken for Kymatio and our implementation to compute the scattering coefficients of images with $L = 8$ and varying values of J on a CPU. Each data point represents an average of 10 runs, with each run processing a batch of 64 images. Error bars are omitted, as they are barely visible.



(a) Results for low-resolution images.



(b) Results for high-resolution images.

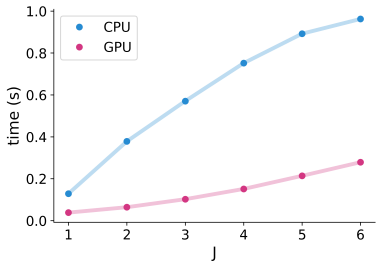
Figure 5.10: Time taken for Kymatio and our implementation to compute the scattering coefficients of images with $L = 8$ and varying values of J on a GPU. Each data point represents an average of 10 runs, with each run processing a batch of 64 images. Error bars are omitted, as they are barely visible.

Kymatio adds a reflection padding to the signal. With these experiments we want to show that our implementation is correct and fast for images with periodic boundary conditions, although it is an unfair comparison with Kyamtio because it enlarges the input signals with padding. It is worth emphasizing that our filter initialization is considerably faster than Kymatio's.

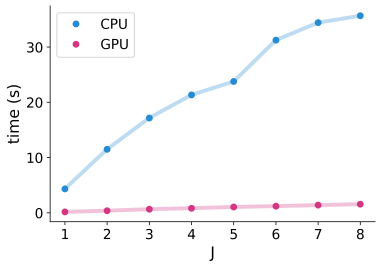
The difference becomes more pronounced with larger values of J, where the filters become bigger, and our method is over 10 times faster than Kymatio. In addition, the time complexity of our method grows linearly with J, while that of Kymatio exhibits an exponential growth. Both methods require similar amounts of memory, as they both traverse the scattering tree in a depth-first manner. This approach reduces memory usage, which is especially important when running the code on a GPU.

5.2.3 Performance on a GPU

Using a GPU instead of a CPU to run Kymatio code can result in significant speedups, especially in the 2-dimensional case. We compare the time taken for Kymatio and our implementation to compute the scattering of low-resolution and high-resolution images in a GPU. The results shown in



(a) Results for low-resolution images.



(b) Results for high-resolution images.

Figure 5.11: Comparison between the time taken for our implementation to compute the scattering coefficients with $L = 8$ and varying values of J on a CPU and on a GPU. Each data point represents an average of 10 runs with each run processing a batch of 64 images. Error bars are omitted, as they are barely visible.

Figure 5.10a and Figure 5.10b show again that our implementation is faster.

Figure 5.11a and Figure 5.11b compare the use of a CPU or a GPU for the particular case of our implementation. As we expected, the vectorization of our code is suited for GPUs and we observe remarkable results across all values of J.

Given these results, we are considering proposing a pull request to modify the filter initialization in Kymatio with a vectorized version in order to further optimize the code's speed.

Synthesis with the Scattering Transform

6

High-dimensional generative models are employed to generate new data that exhibits similar characteristics to a given input dataset. To accomplish this, the models must capture the underlying patterns in the provided examples. Scattering representations can define geometric priors that exploit the available information while providing robust mathematical guarantees. Previous studies have demonstrated their effectiveness as image representations (Oyallon, Zagoruyko et al. 2019; Joan Bruna 2013). In this section, we explore their capability to represent potential vorticity fields.

6.1 Description of the Generative Method

According to the principle of maximum entropy, the probability distribution that better represents the current knowledge of a system is that with the highest entropy. If we have a constraint on a data distribution given by its scattering representation, we can use it as the sufficient statistics of a maximum entropy model. The effectiveness of this approach has been corroborated, as the scattering transform outperforms the power spectrum and other representations with higher-order statistics for texture synthesis tasks (Joan Bruna 2013; Cheng and Ménard 2021).

We use microcanonical gradient descent models obtained by transporting a maximum entropy measure with a gradient descent algorithm (Portilla 2000; Joan Bruna and Stéphane Mallat 2019). In other words, we use a generative model that produces images with the same summary statistics. If those statistics are indeed sufficient, the generated fields will have similar properties to the input image. To do so, we iteratively minimize

$$\frac{1}{2}\|S_J\boldsymbol{x} - \boldsymbol{y}\|_2^2, \tag{6.1}$$

where $\boldsymbol{y}$ is the vector with the target statistics and $\boldsymbol{x}$ is initialized from i.i.d. Gaussian random variables. In our case, we take $\boldsymbol{y}$ to be the scattering representation of the

image we want to recover. This approach converges for an appropriate energy vector as the one we choose (Joan Bruna and Stéphane Mallat 2019).

6.2 Implementation of the Algorithm

The optimization problem is solved with a gradient descent approach that uses PyTorch's automatic differentiation engine. We choose the Adam optimizer because it has several advantages over other optimization algorithms. Adam is an adaptive optimization algorithm that adjusts the learning rate automatically during training, leading to faster and more robust convergence than traditional stochastic gradient descent. In fact, it combines the benefits of two other optimization techniques, adaptive gradient and root mean square propagation. Moreover, it is computationally efficient, requires minimal memory and little tuning of the hyperparameters (Kingma and Ba 2015).

The optimization is initialized on random Gaussian noise with the mean μ and standard deviation σ of the original image for convenience. As the potential vorticity fields we want to recreate are Gaussian, this is equivalent to normalizing the images beforehand and sampling the random noise from a standard Gaussian distribution. After approximately 100 iterations or steps of the optimizer, the algorithm reaches a plateau in the loss function and produces visually plausible estimates.

We have tuned the hyperparameters of the method to optimize its performance. To ensure convergence, we perform 300 steps of the optimizer for the 64-by-64 images, but fewer steps can be used to reduce resource usage and time without affecting the results. For the 256-by-256 images, we perform 500 steps. The learning rate is set to 10^{-3} for the 64-by-64 images and to 10^{-2} for the 256-by-256 images, after assessing the results obtained with learning rates $10^{-\alpha}$ for integer α with $0 \leq \alpha \leq 7$.

Furthermore, we have explored a variation of the generative algorithm to determine if it can enhance the procedure. We clip the image at each iteration by clamping all its pixels into the range $[\mu - 3\sigma, \mu + 3\sigma]$, but this did not produce any noticeable effect. We conclude that the initial random noise

and the scattering coefficients suffice for the algorithm to correctly scale the image.

The code can be found in the scattering repository.

6.3 Avoiding Aliasing in Discrete Images

After generating some image samples, we noticed that the low-resolution generated images displayed a checkerboard pattern. This pattern arises when adjacent pixels have alternating brightness values. Some of these artifacts are framed in the middle image of Figure 6.2.

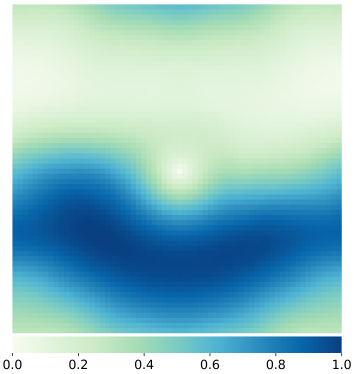


Figure 6.1: Normalized sum of the modulus of the Fourier transform of the filters ψ used for the scattering transform of a 64-by-64 image with $J = 1$ and $L = 4$.

The checkerboard artifacts can be attributed to the ψ_λ filters in the scattering transform failing to capture certain high-frequency components of the signal. Upon examining the frequency coverage of the ψ_λ filters in Figure 6.1, we can see that the outer region of the circle inscribed in the image frequency square is incompletely covered. The lack of enough information of these frequencies in the scattering representation leads to the generation of the artificial patterns in the image. The checkerboard pattern is characterized by frequencies within that particular range of values.

To address this issue, one approach is to remove the unwanted frequency components in the output image with a circular mask. The non-filtered generated image has non-zero values outside the frequency circle as shown in Figure 6.5, and the rightmost image in Figure 6.2 shows that the filter effectively removes these artifacts. This approach is justified by the fact that the images have negligible energy outside the frequency circle.

In natural images, the frequency content is predominantly concentrated within a circle inscribed in the frequency square. However, due to the filtering in the numerical method, the frequency content of potential vorticity fields is further concentrated within a smaller circle centered at the origin. By applying a mask that excludes the artificial contributions outside of this circle, we obtain better reconstructions.

The frequency content distributions and their cumulative versions can be observed in Figure 6.3 and Figure 6.4. For 64-by-64 images, approximately 99.99% of the content is

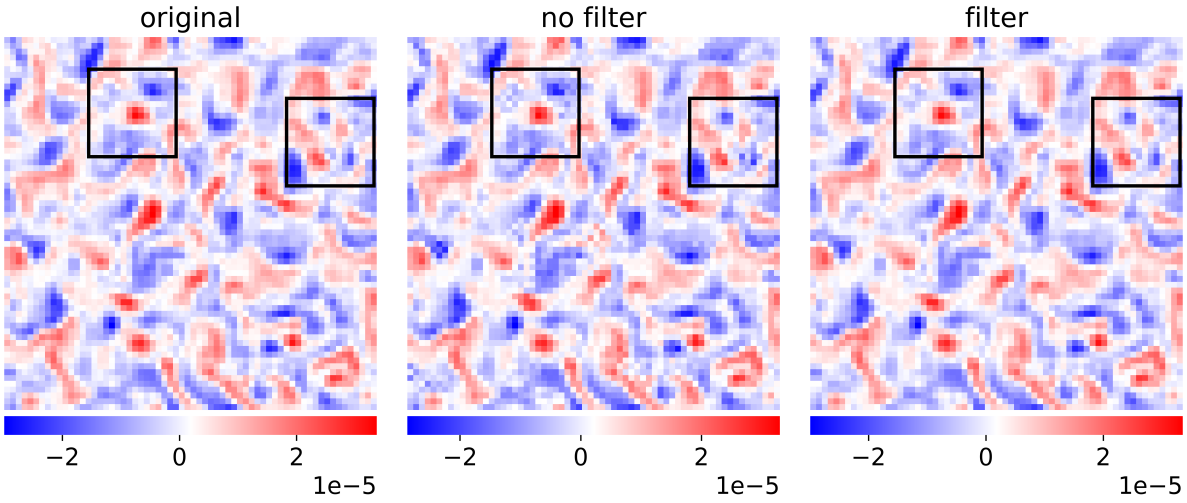


Figure 6.2: Original and generated images from a scattering representation with $J = 1$ and $L = 4$. The checkerboard artifacts were visible in the non-filtered sample and were framed in each image for easy identification.

contained within wavenumbers less than or equal to 26. Similarly, for 256-by-256 images, wavenumbers less than or equal to 100 account for roughly 99.99% of the content. We are conservative in the choice of the radius to not lose relevant frequencies.

6.4 Reconstruction of Potential Vorticity Fields

Figure 6.8 and Figure 6.9 present the results obtained from the generative models for all the possible values of J and $L = 8$. Based on these results, we can conclude that selecting no padding is suitable since there are no visible artifacts along the borders of the generated images, which would be apparent for a bad padding choice.

As detailed in Table Table 5.2, for smaller values of J the size of the scattering representation is comparable to the size of the image. However, as J increases, the size of the scattering transform becomes significantly smaller relative to the image size. Furthermore, as J increases, the averaging effect within the scattering transform becomes more pronounced. This leads to an underconstrained system where very different images end up having the same coefficients. Consequently, the entropy of the model increases with J and the representations obtained capture statistical properties that have demonstrated remarkable performance as

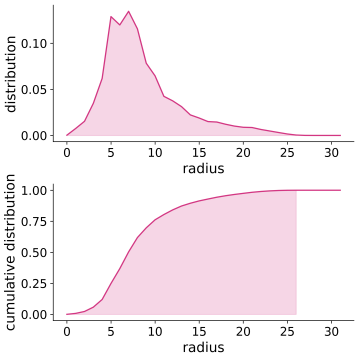


Figure 6.3: Frequency content distribution and cumulative distribution for 64-by-64 potential vorticity fields. These plots represent the averaging of 1000 images of independent first-layer fields and 1000 images of independent second-layer fields. The shadowed area indicates wavenumbers smaller or equal to 26.

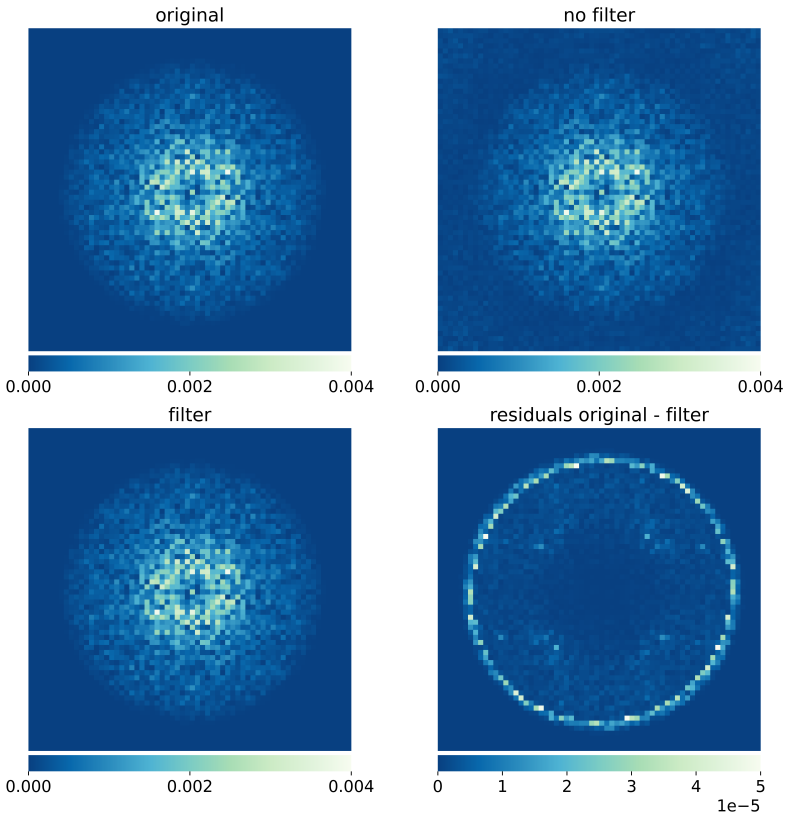


Figure 6.5: Modulus of the Fourier transform of the images in Figure 6.2 and modulus of the Fourier transform of the residuals between the filtered and the original image.

models for textures and stochastic processes, as shown in (Joan Bruna 2013; Sifre and S. Mallat 2013).

Finding the right balance for the scale parameter J is crucial. On one hand, we prefer a larger J because we want to include typical samples from the data distribution in the representation preimage. On the other hand, excessively large J may introduce excessive variability, resulting in the generation of atypical samples for our desired distribution.

The reconstructions exhibit two distinct regimes based on the value of J. In the case of low-resolution images, a small value of J (specifically, $J \leq 3$) defines a scattering representation that captures local information and is nearly invertible and the reconstruction has good perceptual quality. However, when $J \geq 4$, the morphological information in the representation drops, leading to a loss of spatial localization in the reconstructed structures. Nevertheless, the reconstructed images still preserve similar texture details as the originals, and they resemble potential vorticity fields.

Similar trends are observed for high-resolution fields, with the cutoff scale occurring at $J = 4$. Beyond this point, the loss

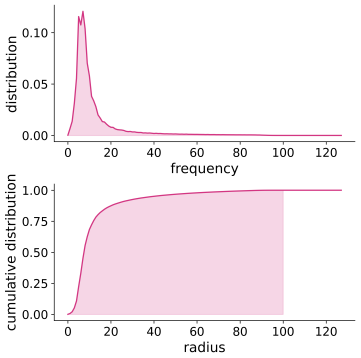


Figure 6.4: Frequency content analysis similar to Figure 6.3 for 256-by-256 potential vorticity fields.

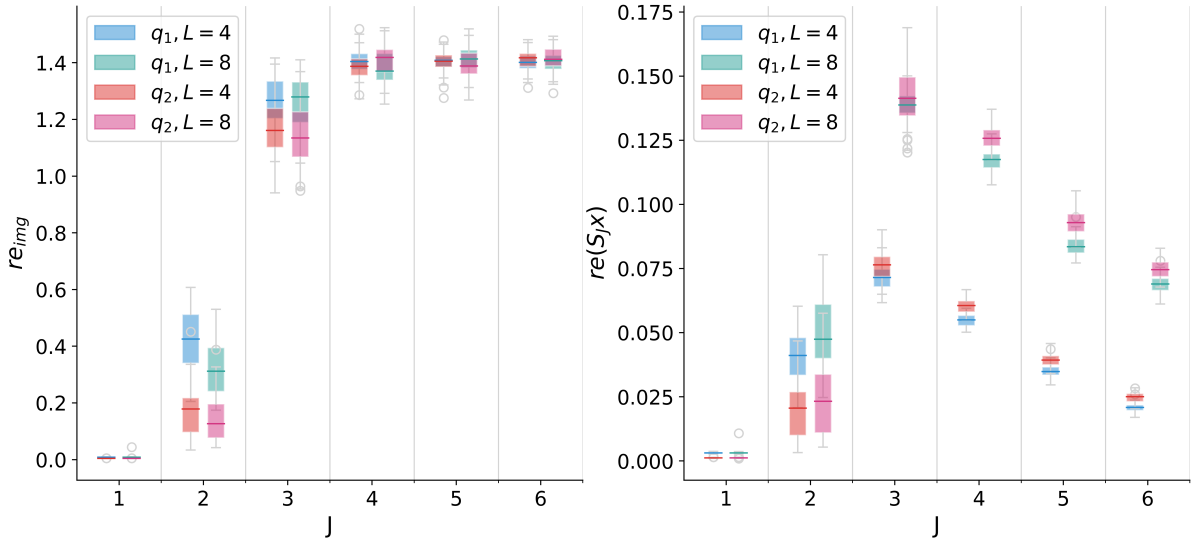


Figure 6.6: Relative errors in the image and scattering domain for varying values of J and L. The box plots depict the performance on 100 low-resolution potential vorticity fields images, divided equally between the top and bottom layers.

of spatial localization becomes evident in the reconstructed images.

Figure 6.6 and Figure 6.7 show the errors obtained by the generative model in both the image domain and the scattering domain, considering various choices of J and L. The relative errors are calculated using the following definitions:

$$re_{img}(x) = \frac{\|x - x_{true}\|_2}{\|x_{true}\|_2} \qquad re(S_J x) = \frac{\|S_J x - S_J x_{true}\|_2}{\|S_J x_{true}\|_2}, \tag{6.2}$$

where $re_{img}(x)$ represents the relative error in the image domain, and $re(S_J x)$ denotes the relative error in the scattering domain.

We observe consistent trends across both low-resolution and high-resolution scenarios. As the value of J increases, the error in the image domain also increases, as the ability to recover the spatial localization of the features is lost by the averaging. In the scattering domain, the error initially rises but then begins to decrease, which is probably related to the changes in the size of the representation. Regarding the different values of L, slightly better results are obtained in the image domain when $L = 8$, as it considers more rotations of the wavelets. In the scattering domain, we cannot compare the errors directly as the representation with $L = 4$ is smaller.

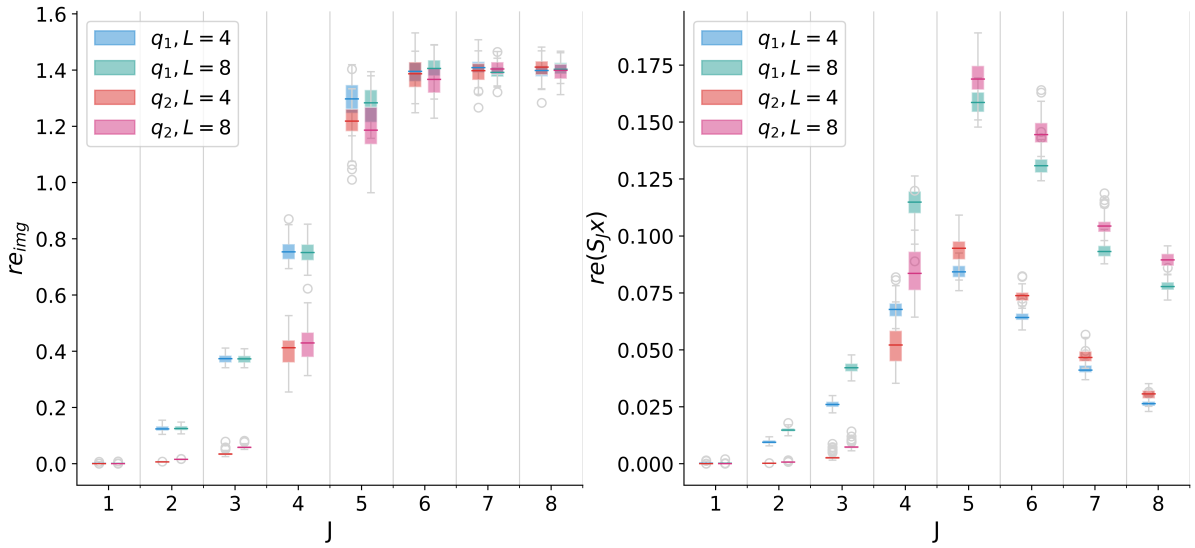
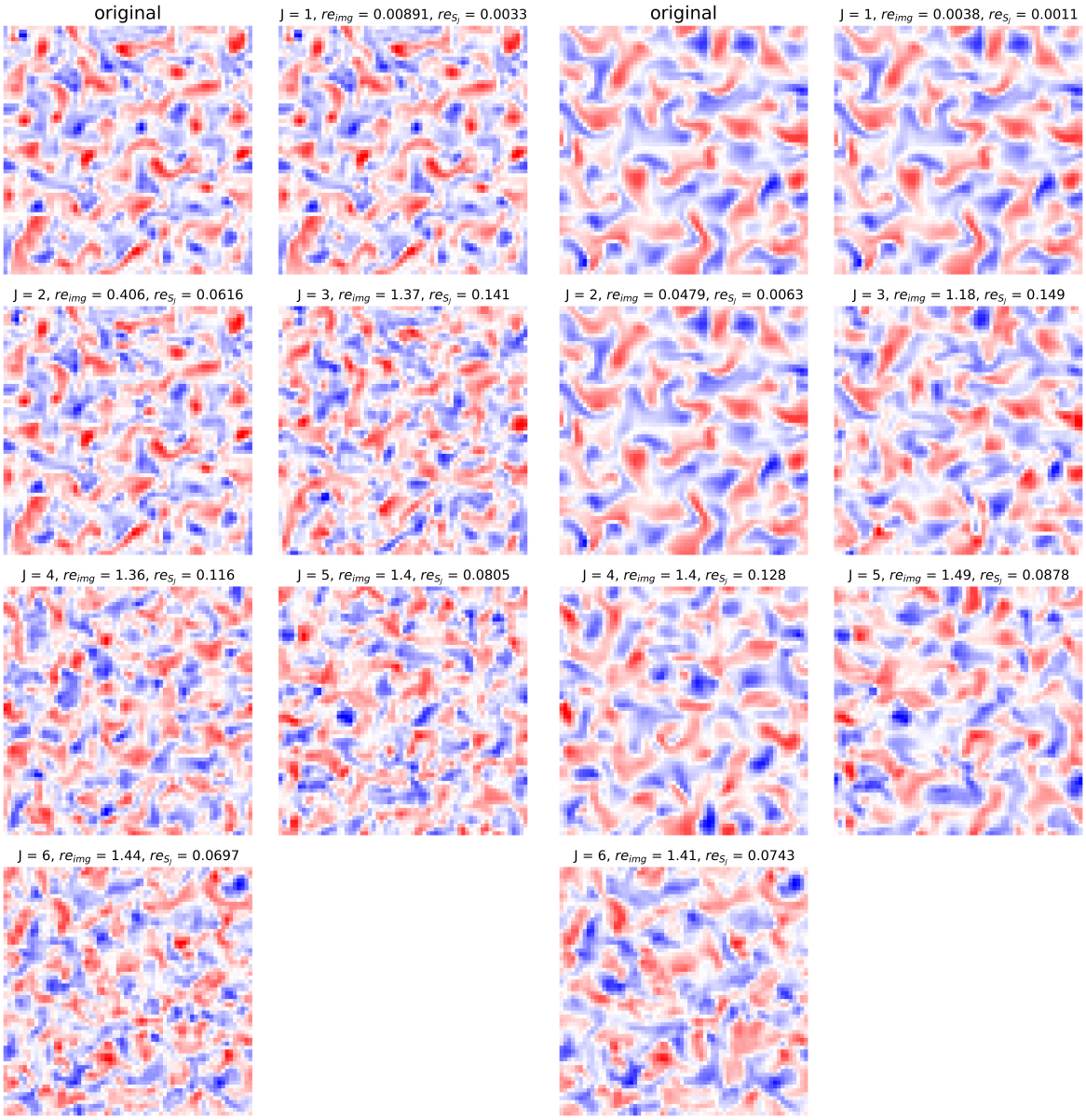


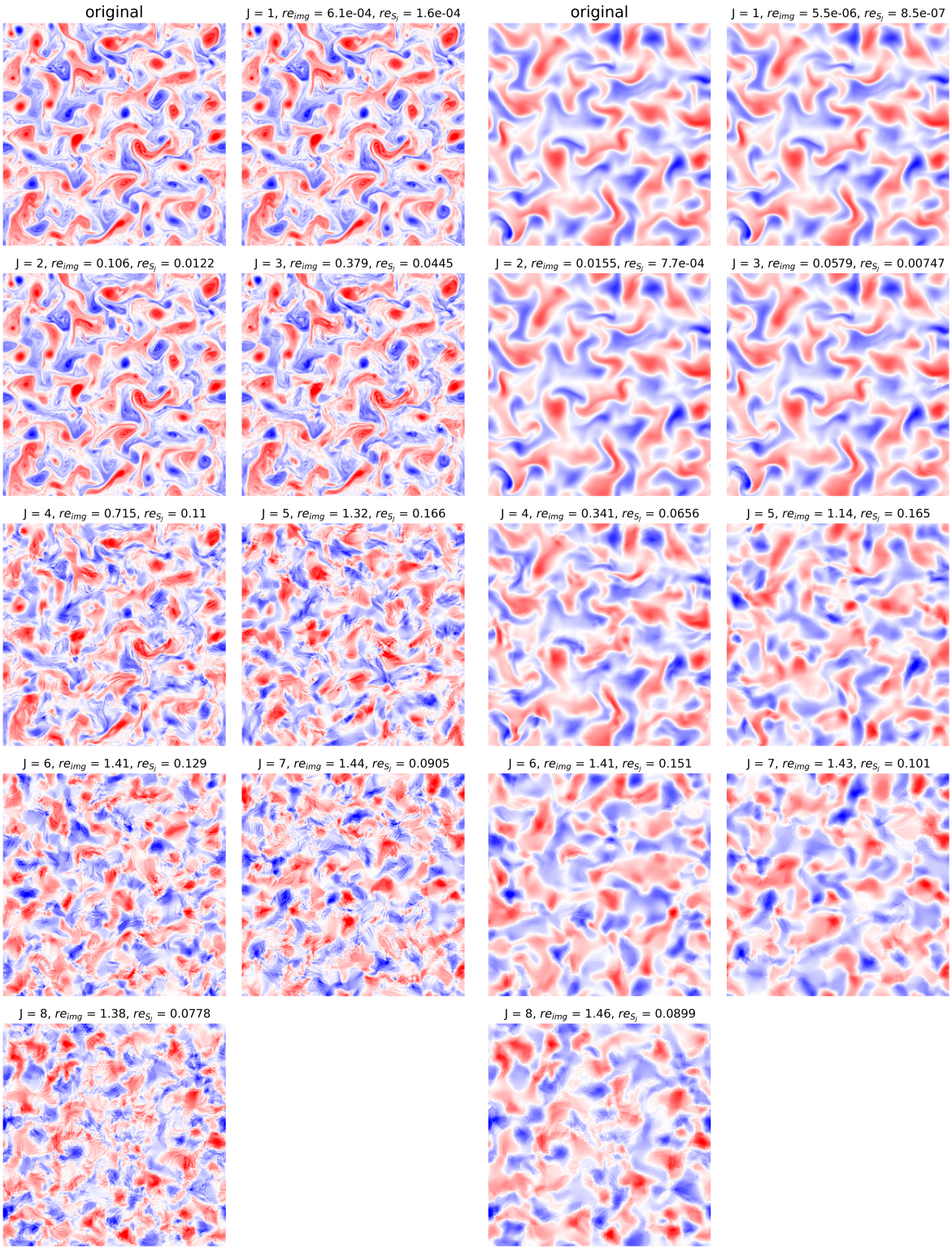
Figure 6.7: Relative errors in the image and scattering domain for varying values of J and L. The box plots depict the performance on 100 high-resolution potential vorticity fields images, divided equally between the top and bottom layers.

It is worth mentioning that further improvement in the recovery of the scattering transform could be achieved by tuning the hyperparameters of the optimization method. However, considering the balance between good recovery and computational time, we are satisfied with the current trade-off. Finally, the presence of smooth features in the bottom layer of the ocean facilitates the modeling with the scattering transform, resulting in better overall performance in the image domain.



(a) Top layer fields (q_1). **(b)** Bottom layer fields (q_2).

Figure 6.8: Reconstruction of potential vorticity 64-by-64 fields from their scattering coefficients for different scale factors J and L set to 8. re_{img} represents the relative error in the image domain and re_{S_J} represents the relative error in the scattering domain.



(a) Top layer fields (q_1). **(b)** Bottom layer fields (q_2).

Figure 6.9: Reconstruction of potential vorticity 256-by-256 fields from their scattering coefficients for different scale factors J and L set to 8. re_{img} represents the relative error in the image domain and re_{S_J} represents the relative error in the scattering domain.

Super-resolution with the Scattering Transform

7

The scattering transform is a powerful tool for regularizing ill-posed inverse problems, such as super-resolution imaging. Super-resolution involves constructing high-resolution images from their downsampled counterparts, typically through models with priors that regularize the estimation.

Recent data-driven approaches, like the use of CNNs, have led to successful results, as training data can help adjust the prior to the empirical distribution (Kim, Lee et al. 2016; Tai, Yang et al. 2017; Huang, He et al. 2017; Tian, Xu et al. 2021). However, multiple valid high-resolution images may correspond to a single low-resolution image and this multimodality can make the mapping between low-resolution and high-resolution signals highly unstable. Consequently, CNNs may suffer from an effect commonly referred to as regression to the mean, resulting in linear blurring in the reconstructed images and, thus, a loss of details and fine textures.

The loss function in CNNs plays a crucial role in guiding the system to identify the regularities and geometric properties of the images. The L^2-norm in the image domain is a common choice, but it is not always correlated with good perceptual quality since it is not stable to small deformations and can lead to the regression to the mean effect. This occurs when the prediction is an average of the modes of a multi-modal distribution, which may not be a typical realization of the distribution. To address this issue and model the multi-modality found in many inverse problems, (Joan Bruna, Sprechmann et al. 2016) proposes modeling the conditional distribution $p(\boldsymbol{x} \mid \boldsymbol{y})$ with a collection of hidden variables that capture its multi-modality, where $\boldsymbol{x} \in \mathbb{R}^M$ represents the high-resolution images and $\boldsymbol{y} \in \mathbb{R}^N$ represents the low-resolution images, with $N < M$.

As proposed in (Joan Bruna, Sprechmann et al. 2016), we use the scattering transform to learn a non-linear representation of the target signal $\Psi(\boldsymbol{x})$ that expresses the multi-modal

distribution as a Gibbs density

$$p(\boldsymbol{x} \mid \boldsymbol{y}) \propto \exp\left(-\|\Psi(\boldsymbol{x}) - \Phi(\boldsymbol{y})\|_2^2\right) \tag{7.1}$$

where $\Psi : \mathbb{R}^M \to \mathbb{R}^K$ and $\Phi : \mathbb{R}^N \to \mathbb{R}^K$ are non-linear mappings to a common high-dimensional space of dimension K.

Our goal is to minimize the uninformative variability of $\boldsymbol{x}$ given $\boldsymbol{y}$ while preserving discriminative information. This can be achieved with the non-linear sufficient statistics $\Psi(\boldsymbol{x})$ given by the scattering coefficients, as they are stable to small geometric deformations and preserve high-frequency information. Furthermore, they do not have high variance because they are computed with a non-expansive operator, as proved in Proposition 4.3.3.

7.1 Description of the Super-Resolution Method

Consider the problem of estimating an unknown image $\boldsymbol{x} \in \mathbb{R}^M$ from a noisy and under-determined signal $\boldsymbol{y} = f(\boldsymbol{x}) + \boldsymbol{\varepsilon} \in \mathbb{R}^N$, where f may be a non-linear map and $\boldsymbol{\varepsilon}$ models additive Gaussian noise. In the super-resolution problem, f is a non-invertible downsampling operator. Consider a Bayesian approach given by the maximum a posteriori (MAP) estimate

$$\begin{aligned} \arg\max_{\boldsymbol{x}} p(\boldsymbol{x} \mid \boldsymbol{y}) &= \arg\max_{\boldsymbol{x}} p(\boldsymbol{y} \mid \boldsymbol{x}) \cdot p(\boldsymbol{x}) && (7.2) \\ &= \arg\max_{\boldsymbol{x}} \log p(\boldsymbol{x}) + \log p(\boldsymbol{y} \mid \boldsymbol{x}), && (7.3) \end{aligned}$$

where we have used Bayes' theorem and the fact that the logarithm is a monotonically increasing function. Under the Gaussian noise assumption, $\log p(\boldsymbol{y} \mid \boldsymbol{x}) \propto -\|\boldsymbol{y} - f(\boldsymbol{x})\|_2^2$ with a positive proportionality constant. The problem can then be reformulated as

$$\arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - f(\boldsymbol{x})\|_2^2 - \alpha \log p(\boldsymbol{x}), \tag{7.4}$$

for $\alpha > 0$. We select a prior $\log p(\boldsymbol{x}) \propto -\|\Psi(\boldsymbol{x}) - \Phi(\boldsymbol{y})\|_2^2$ with a positive proportionality constant. The final formula-

tion of the problem becomes

$$\arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - f(\boldsymbol{x})\|_2^2 - \alpha\|\Psi(\boldsymbol{x}) - \Phi(\boldsymbol{y})\|_2^2. \qquad (7.5)$$

Ψ computes a scattering representation of the residuals, whereas Φ serves as a predictive model for the scattering representation of the downsampled image. We train this model by minimizing $\|\Psi(\boldsymbol{x}) - \Phi(\boldsymbol{y})\|_2^2$ on the parameters of Φ with a dataset of pairs $(\boldsymbol{x}_i, \boldsymbol{y}_i)_{i\in\mathcal{I}}$.

Given the non-linear nature of the scattering transform, we explore $\Phi(\boldsymbol{y}) = M\varphi(\boldsymbol{y})$, where φ is a scattering representation of $\boldsymbol{y}$ and M is a linear operator learned through least-squares regression. We hope that the non-linearity in the representation allows for a linear model in the scattering domain that hallucinates the high frequencies. Specifically, we choose the representation φ such that

$$J_\Psi - J_\varphi = 2.$$

This ensures that the coefficients of the downsampled image, representing the system's low-frequency components, are a subset of the coefficients of the high-resolution image. Here, J_Ψ and J_φ indicate the scales J of the scattering representations Ψ and φ, respectively.

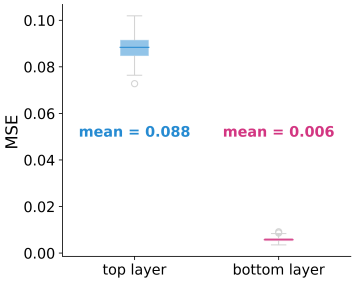


Figure 7.1: MSE for 1000 images from each layer of the ocean obtained with a SRCNN.

7.2 Results using a CNN

First, we apply super-resolution using a standard CNN to visualize the regression to the mean effect in the potential vorticity fields. For this purpose, we utilize the lightweight SRCNN architecture introduced by (Dong, Loy et al. 2014), which achieved state-of-the-art performance in the super-resolution task when it was released. Although there have been subsequent advancements in architectures that yield better results, we opt for SRCNN due to its simple design and still good performance. The CNN consists of three layers with $64, 32$ feature maps and filter sizes of 9×9, 5×5, and 5×5, respectively, and ReLU non-linearities applied at the first two layers. We have trained two CNN with weight decay, one for the top-layer and the other for the bottom layer of the 2-layer quasi-geostrophic model, and they are trained independently with a dataset of 60,000 images. Each

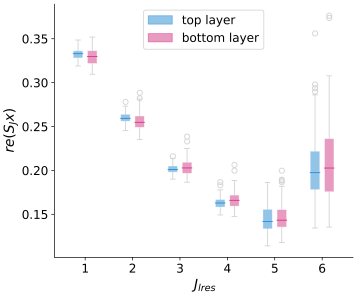


Figure 7.2: Relative error of the reconstruction of the high-resolution scattering coefficients for 1000 images from each layer of the ocean obtained with a linear operator in the scattering domain.

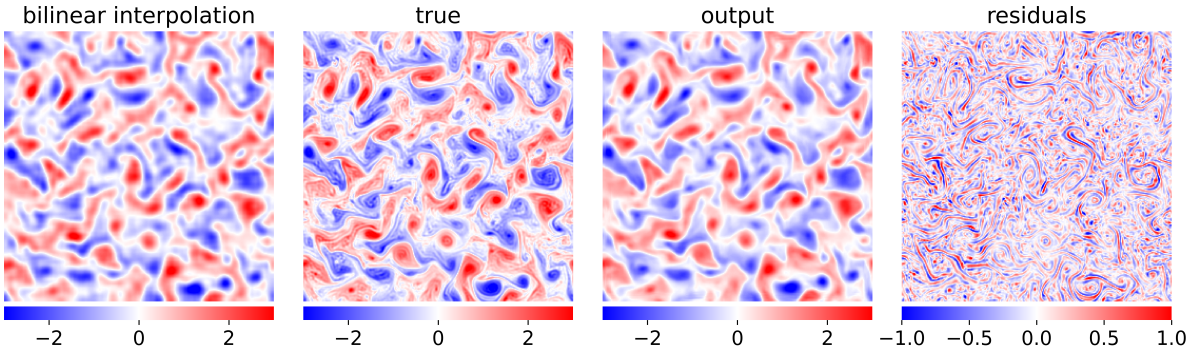


Figure 7.3: Super-resolved top-layer potential vorticity field using a SRCNN.

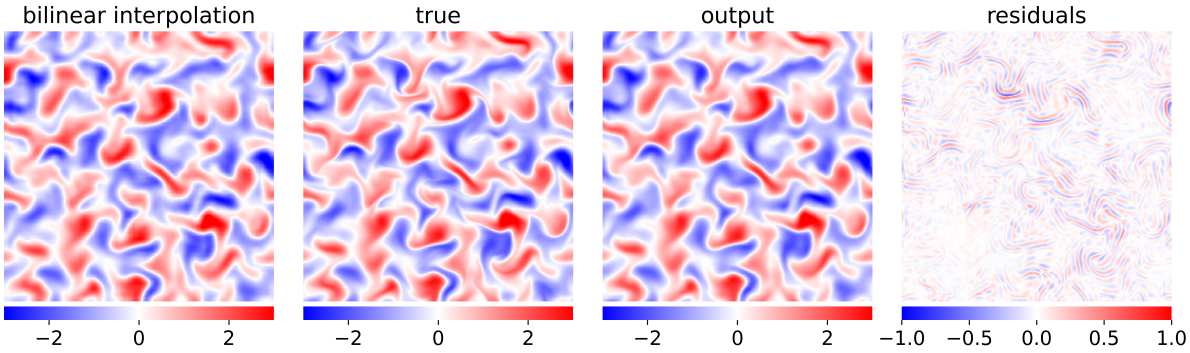


Figure 7.4: Super-resolved bottom-layer potential vorticity field using a SRCNN.

image represents the final state of a trajectory with different initial conditions, ensuring their independence.

In Figure 7.1, we observe that the bottom layer exhibits the best results. This can be attributed to its smoother features, which are less affected by the linear blurring induced by the regression to the mean effect. Figure 7.3 and Figure 7.4 present the results for two instances of potential vorticity fields. Notably, the CNN outputs resemble the bilinear interpolation fields, which are obtained by upsampling the low-resolution images. The CNN predicts the average of the modes of the multi-modal distribution of high-resolution fields, which leads to the loss of filaments in the outputs.

7.3 Results using the Scattering Transform

In the case of the results obtained using the scattering transform, we observe in Figure 7.5 and Figure 7.6 that the linear operator used in the scattering domain has the capability to reconstruct the high-frequencies contained within the scattering coefficients. This leads to significantly

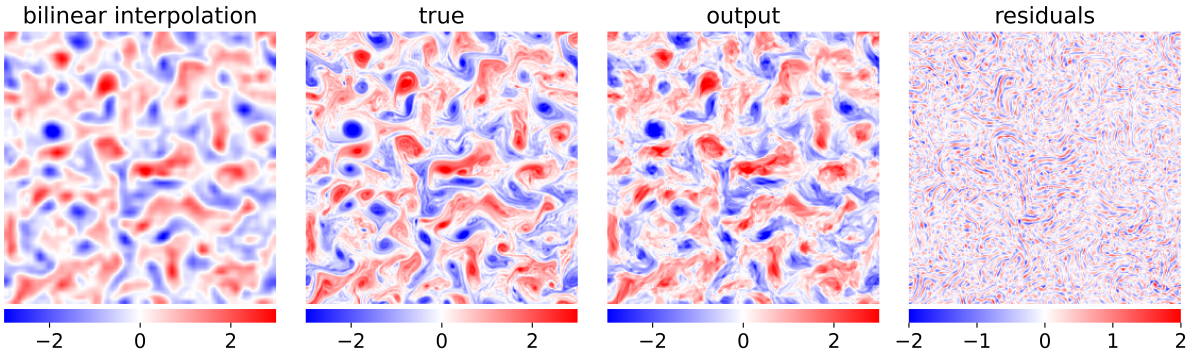


Figure 7.5: Super-resolved bottom-layer potential vorticity field using a linear operator in the scattering domain for $J = 1$ and $L = 4$.

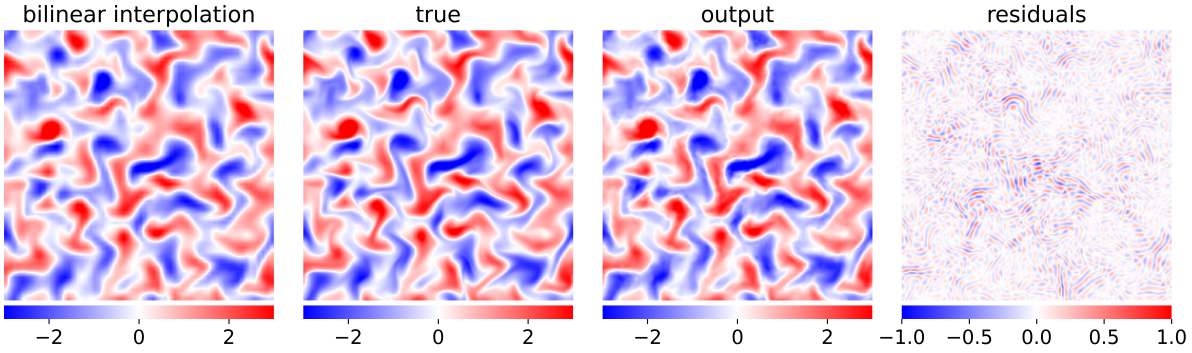


Figure 7.6: Super-resolved bottom-layer potential vorticity field using a linear operator in the scattering domain for $J = 1$ and $L = 4$.

improved perceptual quality results compared to those achieved using a CNN. This is promising and holds potential for the modeling of subgrid forcing terms.

Figure 7.2 demonstrates the effectiveness of the linear operator in reconstructing the coefficients. While the results are already quite satisfactory, with the coefficients being recovered with a relative error of 10-30% and resulting in perceptually high-quality synthesized fields, we can explore the possibility of improving this performance through the use of non-linear regression in the scattering domain. Despite the advantages of the linear operator, such as faster operations and better integration with a climate model, investigating the benefits offered by a non-linear approach would be valuable.

8 Discussion and Future Directions

In conclusion, our thesis has demonstrated that the scattering transform holds promise as an approach to enhance existing models for subgrid forcing terms by serving as a potential regularizer. Through our modeling study, we have shown that the scattering transform effectively encodes crucial features of the potential vorticity fields.

One of the primary challenges with data-driven methods is that the error observed point by point does not guarantee generalizability, and often the introduction of data-driven subgrid forcing can destabilize numerical methods. To address this issue, it is essential to incorporate physical priors into the loss functions to guide the model towards physically-informed solutions. By incorporating these priors in the scattering domain, we can mitigate the regression to the mean effect commonly encountered in CNNs, which are the primary architecture used for subgrid predictions in deep learning. In the near future, we intend to integrate scattering regularizers into existing CNNs to determine if they yield improvements in the models' performance.

Furthermore, we aim to investigate how the time stepping process influences the scattering coefficients. This can be approached in two ways. Firstly, employing a high-dimensional model such as a deep learning approach, we can update the scattering coefficients and perform time stepping in the scattering domain, reconstructing the fields from this representation. Alternatively, due to the analytic nature of the scattering coefficients, we can explore the governing equations in the scattering domain to identify more suitable wavelet choices that better capture the important statistics specific to ocean modeling problems.

While our results have been presented solely in the context of the 2-layer quasi-geostrophic model, it is important to assess the generalizability of our findings to other models and regimes. After completing the modeling phase, it is necessary to implement subgrid parametrizations that employ the scattering transform and conduct offline and online tests to evaluate their performance and ascertain any potential improvements.

In summary, this thesis has shed light on the promising capabilities of the scattering transform as a regularizer for improving existing models for subgrid forcing terms. By incorporating physical priors and exploring the time stepping process and different scattering representations, we aim to further enhance our understanding and application of the scattering transform in the field of ocean modeling.

References

Le Treut, H., R. Somerville et al. (2007). 'Historical Overview of Climate Change'. In: *Climate Change 2007: The Physical Science Basis. Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change*. Ed. by S. Solomon, D. Qin et al. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA. Chap. 1 (cit. on p. 3).

Eltahir, Elfatih and Aaron Krol (2021). *Climate models*. URL: https://climate.mit.edu/explainers/climate-models (visited on 22/03/2023) (cit. on p. 4).

Allen, Jesse and NASA Earth Observatory (2011). *The Eddy and the Plankton*. URL: https://earthobservatory.nasa.gov/images/77120/the-eddy-and-the-plankton (visited on 23/04/2023) (cit. on p. 4).

– (2016). *A Day in the East Korea Current*. URL: https://earthobservatory.nasa.gov/images/88235/a-day-in-the-east-korea-current (visited on 23/04/2023) (cit. on p. 4).

Legg, Sonya and Robert Hallberg (2023). *Ocean Mixing*. URL: https://www.gfdl.noaa.gov/ocean-mixing/ (visited on 23/04/2023) (cit. on p. 4).

Adcroft, Alistair, Stephen Griffies et al. (2023). *Ocean Mesoscale Eddies*. URL: https://www.gfdl.noaa.gov/ocean-mesoscale-eddies/ (visited on 22/03/2023) (cit. on p. 4).

Grotjahn, R. (2003). 'BAROCLINIC INSTABILITY'. In: *Encyclopedia of Atmospheric Sciences*. Elsevier, pp. 179–188. DOI: 10.1016/b0-12-227090-8/00076-2 (cit. on p. 5).

Smith, David C., J. H. Morison et al. (1984). 'Topographic generation of an eddy at the edge of the East Greenland Current'. In: *Journal of Geophysical Research* 89.C5, p. 8205. DOI: 10.1029/jc089ic05p08205 (cit. on p. 5).

NOAA (2023). *What is the Atlantic Meridional Overturning Circulation (AMOC)?* URL: https://oceanservice.noaa.gov/facts/amoc.html (visited on 24/04/2023) (cit. on p. 5).

Zhang, Rong (2023). *Large-scale Ocean Dynamics*. URL: https://www.gfdl.noaa.gov/large-scale-ocean-dynamics/ (visited on 23/04/2023) (cit. on p. 5).

Vallis, Geoffrey K. (2017). *Atmospheric and Oceanic Fluid Dynamics: Fundamentals and Large-Scale Circulation*. 2nd ed. Cambridge University Press (cit. on p. 6).

Mechoso, C.R. and A. Arakawa (2015). 'NUMERICAL MODELS | General Circulation Models'. In: *Encyclopedia of Atmospheric Sciences*. Elsevier, pp. 153–160. DOI: 10.1016/b978-0-12-382225-3.00157-2 (cit. on p. 6).

Pedlosky, Joseph (1970). 'Finite-Amplitude Baroclinic Waves'. In: *Journal of Atmospheric Sciences* 27.1, pp. 15–30. DOI: https://doi.org/10.1175/1520-0469(1970)027<0015:FABW>2.0.CO;2 (cit. on p. 7).

Abernathey, Ryan, rochanotes et al. (2022). *pyqg/pyqg: v0.7.2*. Version v0.7.2. DOI: 10.5281/zenodo.6563667 (cit. on pp. 7, 8).

Ross, Andrew, Ziwei Li et al. (2023). 'Benchmarking of Machine Learning Ocean Subgrid Parameterizations in an Idealized Model'. In: *Journal of Advances in Modeling Earth Systems* 15.1. DOI: 10.1029/2022ms003258 (cit. on pp. 8, 12, 14).

Chen, Amy, William Barham et al. (2018). 'Comparing Eddy-Permitting Ocean Model Parameterizations via Lagrangian Particle Statistics in a Quasigeostrophic Setting'. In:

Journal of Geophysical Research: Oceans 123.8, pp. 5637–5651. DOI: 10.1029/2018jc014182 (cit. on p. 9).
Lesieur, Marcel (1997). *Turbulence in Fluids*. Springer Netherlands (cit. on p. 10).
Sagaut, Pierre (2001). *Large Eddy Simulation for Incompressible Flows*. Springer Berlin Heidelberg (cit. on pp. 10, 11).
Yuval, Janni and Paul A. O'Gorman (2020). 'Stable machine-learning parameterization of subgrid processes for climate modeling at a range of resolutions'. In: *Nature Communications* 11.1. DOI: 10.1038/s41467-020-17142-3 (cit. on pp. 14, 15).
Guan, Yifei, Adam Subel et al. (2023). 'Learning physics-constrained subgrid-scale closures in the small-data regime for stable and accurate LES'. In: *Physica D: Nonlinear Phenomena* 443, p. 133568. DOI: 10.1016/j.physd.2022.133568 (cit. on p. 14).
Rackauckas, Christopher, Yingbo Ma et al. (2020). *Universal Differential Equations for Scientific Machine Learning*. DOI: 10.48550/ARXIV.2001.04385. URL: https://arxiv.org/abs/2001.04385 (cit. on p. 14).
Shankar, Varun, Vedant Puri et al. (2023). 'Differentiable physics-enabled closure modeling for Burgers' turbulence'. In: *Machine Learning: Science and Technology* 4.1, p. 015017. DOI: 10.1088/2632-2153/acb19c (cit. on p. 14).
Mallat, S. (2012). 'Group Invariant Scattering'. In: *Communications on Pure and Applied Mathematics* 65.10, pp. 1331–1398. DOI: https://doi.org/10.1002/cpa.21413 (cit. on pp. 16, 26, 27).
Waldspurger, I. (2017). 'Exponential decay of scattering coefficients'. In: *2017 International Conference on Sampling Theory and Applications (SampTA)*. IEEE. DOI: 10.1109/sampta.2017.8024473 (cit. on pp. 16, 25).
Bruna, J. (2022). 'The Scattering Transform'. In: *Mathematical Aspects of Deep Learning*. Cambridge University Press, pp. 338–399. DOI: 10.1017/9781009025096.009 (cit. on p. 16).
Mallat, S. and S. Zhong (1992). 'Characterization of signals from multiscale edges'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.7, pp. 710–732. DOI: 10.1109/34.142909 (cit. on p. 21).
Andreux, Mathieu, Tomás Angles et al. (2020). 'Kymatio: Scattering Transforms in Python'. In: *Journal of Machine Learning Research* 21.60, pp. 1–6 (cit. on p. 28).
Oyallon, Edouard, Sergey Zagoruyko et al. (2019). 'Scattering Networks for Hybrid Representation Learning'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.9, pp. 2208–2221. DOI: 10.1109/tpami.2018.2855738 (cit. on pp. 32, 36).
Bruna, Joan (2013). 'Scattering Representations for Recognition'. Déposée Novembre 2012. Theses. Ecole Polytechnique X (cit. on pp. 36, 40).
Cheng, Sihao and Brice Ménard (2021). *How to quantify fields or textures? A guide to the scattering transform* (cit. on p. 36).
Portilla, Javier (2000). In: *International Journal of Computer Vision* 40.1, pp. 49–70. DOI: 10.1023/a:1026553619983 (cit. on p. 36).
Bruna, Joan and Stéphane Mallat (2019). 'Multiscale sparse microcanonical models'. In: *Mathematical Statistics and Learning* 1.3, pp. 257–315. DOI: 10.4171/msl/7 (cit. on pp. 36, 37).

Kingma, Diederik P. and Jimmy Ba (2015). 'Adam: A Method for Stochastic Optimization'. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun (cit. on p. 37).

Sifre, L. and S. Mallat (2013). 'Rotation, Scaling and Deformation Invariant Scattering for Texture Discrimination'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 40).

Kim, Jiwon, Jung Kwon Lee et al. (2016). 'Accurate Image Super-Resolution Using Very Deep Convolutional Networks'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 45).

Tai, Ying, Jian Yang et al. (2017). 'Image Super-Resolution via Deep Recursive Residual Network'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 45).

Huang, Huaibo, Ran He et al. (2017). 'Wavelet-SRNet: A Wavelet-Based CNN for Multi-Scale Face Super Resolution'. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (cit. on p. 45).

Tian, Chunwei, Yong Xu et al. (2021). 'Coarse-to-Fine CNN for Image Super-Resolution'. In: *IEEE Transactions on Multimedia* 23, pp. 1489–1502. DOI: 10.1109/tmm.2020.2999182 (cit. on p. 45).

Bruna, Joan, Pablo Sprechmann et al. (2016). 'Super-Resolution with Deep Convolutional Sufficient Statistics'. In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun (cit. on p. 45).

Dong, Chao, Chen Change Loy et al. (2014). 'Learning a Deep Convolutional Network for Image Super-Resolution'. In: *Computer Vision – ECCV 2014*. Springer International Publishing, pp. 184–199. DOI: 10.1007/978-3-319-10593-2_13 (cit. on p. 47).