Ocean parameterizations in an idealized model using machine learning

Auteur : Mangeleer, Victor
Promoteur(s) : Louppe, Gilles
Faculté : Faculté des Sciences appliquées
Diplôme : Master en ingénieur civil physicien, à finalité approfondie
Année académique : 2022-2023
URI/URL : http://hdl.handle.net/2268.2/18251

Ocean subgrid parameterization

in an idealized model using machine learning

Author:
Victor Mangeleer

Supervisor:
Gilles Louppe

Master thesis completed in order to obtain the degree of
Master of Science in Engineering Physics
by Mangeleer Victor

University of Liège
Faculty of Applied Sciences

Academic year 2022–2023

Abstract

This research aims to explore novel methods for parameterizing the contributions of subgrid-scale processes, which refer to physical phenomena occurring at scales finer than the simulation resolution. More precisely, this work is built upon the research of Ross et al., 2023, who, after many years of parameterization development, have created a framework to properly conduct the assessment of the quality of a parameterization.

In addition to replicating their findings, this study extends its scope by attempting to enhance their results through a series of experiments involving more complex datasets. Furthermore, and perhaps most significantly, it delves into the use of Fourier Neural Operators for modeling subgrid-scale process contributions. These neural networks were recently introduced by Li et al., 2020, and have already exhibited impressive results in many areas of computational fluid dynamics. Hence, while building upon the foundation laid by Ross et al., 2023, this study also pioneers the use of Fourier Neural Operators in this context, subjecting them to comprehensive evaluation within the established benchmarking framework.

In conclusion, this research not only facilitates a comprehensive grasp of the underlying physics in ocean-climate simulations but also delves into unexplored realms by leveraging state-of-the-art deep learning techniques for modeling subgrid-scale processes contributions. The conclusive results show promise and underscore the notion that the most captivating discoveries frequently emerge at the crossroads of two captivating scientific domains.

Statement of authorship

I, Victor Mangeleer, born on August 16, 2000, hereby confirm that I am the author of the master's thesis titled "*Ocean Subgrid Parameterization in an Idealized Model Using Machine Learning*."

All sources used in my research have been duly acknowledged and referenced. Any figures included in this work have either been handcrafted by me, generated using my original code, or borrowed from a paper with proper citation.

Chapter 2, which provides an introduction to fluid mechanics, has been guided by Prof. Vincent Terrapon's course *Introduction to Computational Fluid Dynamics*. While I have independently written the content, I acknowledge that the chapter draws significant inspiration from the course.

Furthermore, I acknowledge that I have refined some portions of my original text in collaboration with the ChatGPT language model to enhance its clarity and coherence.

Acknowledgements

"*For what's money without hapiness ? It's like a hard time without the people you love.*"

- J. Cole

Every journey has to come to an end eventually. I am truly thankful to the University of Liège for helping me become who I am today.

I would also like to extend my appreciation to my advisor, Prof. Gilles Louppe. I do not think he realizes how much believing in someone can actually change their life for the better. Thanks a lot, and I am really looking forward to more collaborations in the future.

I also want to thank all the professors in the Master of Physics Engineering program at the University. Since we were only a small group of students, we got to know each other well. I have met some amazing people who are all really interesting. Your dedication throughout these last two years of the master's program has made me even more excited about doing research. Your passion for what you do has motivated me more than ever.

Lastly, my deepest thank you goes to my friends, family, and girlfriend. It is not possible to put together words strong enough to explain how grateful I am for having you in my life.

I love you all.

Contents

Chapter One

A call for climate simulation

"What is the use of a house if you haven't got a tolerable planet to put it on ?"

- Henry David Thoreau

1.1 INTRODUCTION

In today's world, the global climate crisis has transcended its status as a mere environmental issue to emerge as a shared concern for humanity's future. The fate of our species is at stake, urging us to face the dire consequences of our actions. Over the past two decades, the Earth has witnessed some of its most devastating climate disasters, including hurricanes (Katrina 2005, Harvey 2017), cyclones (Nargis 2008, Idai 2019), floods (Pakistan 2010, Belgium 2021), heatwaves (Russia 2010, Arctic 2020), bushfires (Australia 2019–2020), and many more. Moreover, as seen in Fig. 1, the alarming trend of identified climate disasters has increased fourfold since 1970.

Even worse, oxygen concentrations in both the open ocean and coastal waters have been declining since at least the middle of the 20th century (Karstensen, Stramma, and Visbeck, 2008; Stanev et al., 2013; Capet et al., 2016; Breitburg et al., 2018). This loss of oxygen, known as deoxygenation, is a significant phenomenon occurring in oceans increasingly damaged by human activities. These activities have led to elevated temperatures, extremely high CO2 levels and nutrient inputs, which, in turn, have affected the population and distribution of marine species.

With just a handful of examples, one can grasp the significance and necessity of climate modeling as it will enable the simulation, understanding, and prediction of the Earth's climate system. It will help to reconstruct past climates, to project future climate scenarios based on various factors like emissions and policies, and to assess the impact of climate change on numerous sectors.

Nevertheless, it is unrealistic to expect an earth-scale numerical simulation that flawlessly captures all the intricacies of global wind circulation and ocean currents. Undoubtedly, the computational cost would be untractable, and by the time the simulation concludes, significant environmental damage could have occurred. Therefore, an effective approach is to conduct simulations at a large spatial scale but with reduced resolution. This balances computational time while sacrificing some details. The challenge lies in efficiently incorporating neglected physical processes, a topic of considerable interest within the scientific community and the focus of this master's thesis.

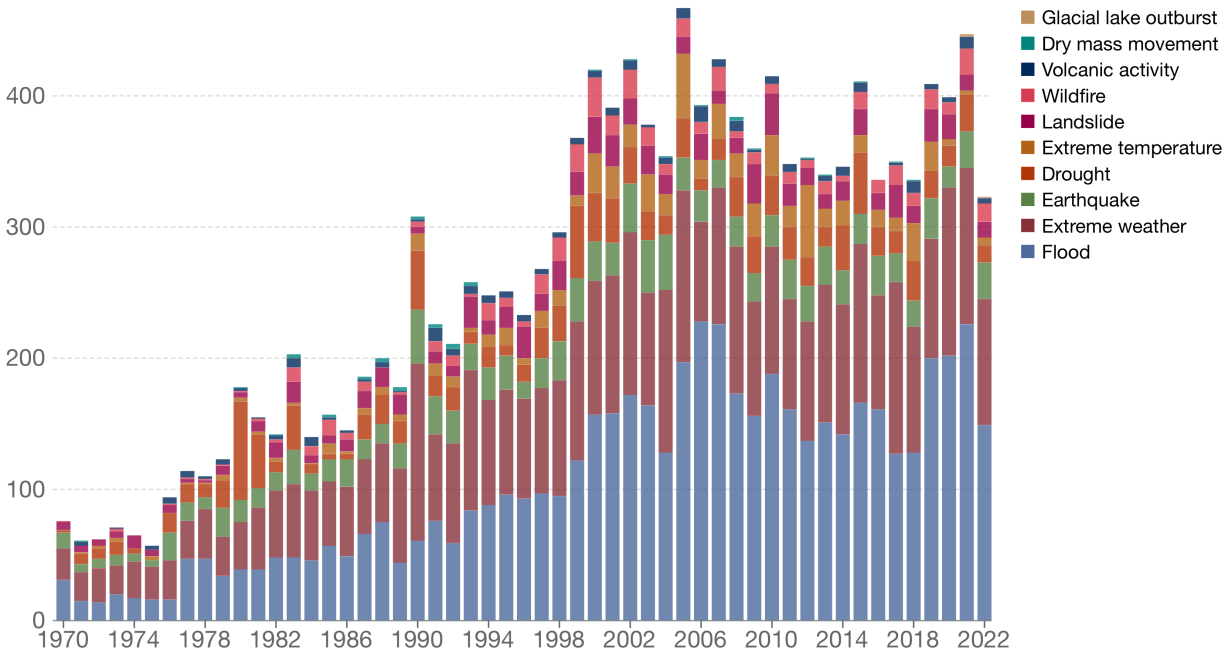


Figure 1: Global reported natural disasters by type from 1970 to 2022. The annual reported number of natural disasters are categorised by type and this includes both weather and non-weather related disasters.

1.2 STATE OF THE ART

In the context of oceans, whether one is studying global wind circulation or ocean currents, simulations frequently involve turbulent flows. They exhibit chaotic, irregular, and random motion of fluid particles, arising from complex interactions between different scales of motion in the fluid. In contrast to laminar flow, where fluid particles move smoothly in parallel layers, ocean flows are distinguished by the presence of vortices, eddies, and fluctuations in velocity and pressure.

Hence, simulations necessitate a considerable scale and high resolution to adequately capture small-scale turbulent structures. However, the associated computational time becomes prohibitive, necessitating a reduction in resolution. Nonetheless, in lower-resolution simulations, smaller turbulent eddies and fluctuations remain unresolved due to grid limitations, resulting in an incomplete portrayal of turbulent processes and subsequently impacting the simulation accuracy.

Nowadays, simulations are commonly conducted at the mesoscale resolution, typically ranging from 10 to 100 kilometers. However, these overlooked physical processes occur at the submesoscale level (less than 10 kilometers). A first question that arises is whether these disregarded processes can be safely overlooked or if their contribution holds significant importance.

In 1941, Andrey Kolmogorov introduced the concept of the Kolmogorov cascade, which elucidates the energy transfer across various scales in turbulent flows until reaching scales where energy dissipates due to viscous effects (Kolmogorov, 1941). Consequently, it is evident that disregarding small-scale physical processes in simulations should impact the energy budget.

This observation has been consistently emphasized in numerous research studies. For instance, Lévy, Resplandy, et al., 2012 demonstrated the significant sensitivity of ocean dynamical solutions to grid resolution. In fact, increasing the simulation resolution to O(1) km resulted in an enhanced energy spectrum for all eddy scales, leading to a complete alteration of flow dynamics, as depicted in Fig.2.

Nevertheless, in certain regions of the globe, mesoscale dynamics have been proven to be the primary driver of biological production by modulating nutrient supplies throughout the year (Resplandy, Lévy, Madec, et al., 2011; Lévy, Ferrari, et al., 2012). However, to accurately predict mesoscale behavior, a correct energy spectrum is necessary, which can only be achieved by resolving submesoscale processes.

In conclusion, the answer to this first question leans towards "no, since these small-scale processes cannot be overlooked" but this topic still remains a subject of intense investigation within the scientific community (Resplandy, Lévy, d'Ovidio, et al., 2009; Lévy and Martin, 2013; Ramachandran, Tandon, and Mahadevan, 2014; Djath et al., 2014; Martin et al., 2015; Su et al., 2018).

Subsequently, the following question arises: How can one account for the missing contributions of these submesoscale processes while achieving faster simulations than high-resolution approaches and accurately representing flow dynamics ? In other words, how can one derive a parameterization for this absent physics ?

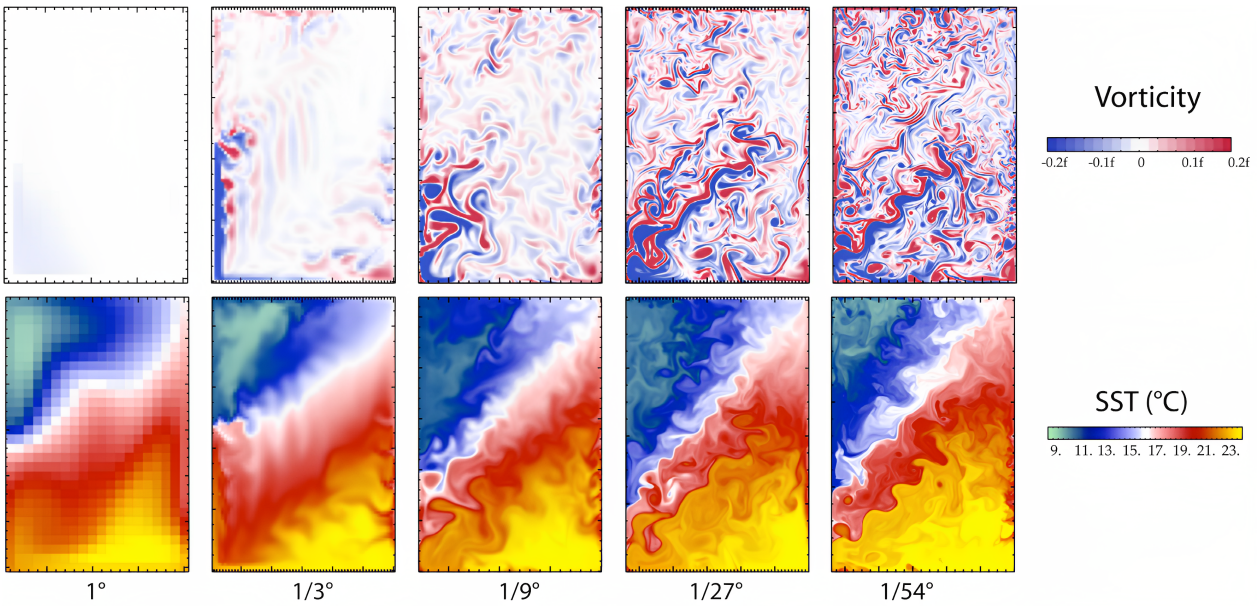


Figure 2: Snapshots of surface relative vorticity and sea-surface temperature (SST) simulated with increasing grid resolution, from 1° (= 111 km) to 1/54° (≈ 2 km) generated by the model of Lévy, Resplandy, et al., 2012

Over the years, many scientists have attempted to address this question. The developement of turbulence closure modelling, i.e. finding a parameterization for the missing contributions, using analytical formulations can be traced back to the mid-20th century.

In 1925, Prandtl published a paper that profoundly influenced the understanding and modeling of turbulence in fluid dynamics (Prandtl, 1925). Through meticulous analysis of experimental data, Prandtl provided essential insights into the nature of turbulence, making it easier to comprehend and predict turbulent flows in practical engineering applications.

Building upon Prandtl's work, many researchers have made significant advancements in fluid mechanics and a perfect example there is Smagorinsky's turbulence model (Smagorinsky, 1963). This model represents an important milestone in turbulence modeling since it is the first to incorporate subgrid-scale modeling in numerical simulations. Its application in various atmospheric and oceanic models, such as weather prediction and climate models, has enabled reasonably efficient simulations of turbulent flows.

Following this, the development of two major turbulence models, the $k - \varepsilon$ (Singhal and Spalding, 1981) and $k - \omega$ (Menter, 1992) models, has been instrumental in simulating turbulent flows and continues to be widely utilized in current research and engineering practices.

Another significant contribution to turbulence modeling is the Backscattering model (Jansen and Held, 2014; Jansen, Held, et al., 2015). Indeed, this model addresses the backscatter mechanism, where energy is transferred from unresolved subgrid scales back to the resolved scales due to interactions between resolved-scale and subgrid-scale motions. Implementing backscatter energy in a physically consistent manner enhances the accuracy of turbulence models, particularly in regions with complex turbulence behavior.

However, all of these analytical parameterizations encounter a common challenge known as the closure problem. Introducing additional equations often involves higher-order statistical moments, leading to an excess of unknowns compared to equations. Furthermore, universality becomes an issue as turbulence behaves differently in various flow regimes, limiting the applicability of the model. Moreover, the validity range of these models is restricted, and they may struggle to accurately predict turbulence behavior outside this range. Additionally, turbulence closure models are sensitive to initial and boundary conditions, and they may lack accuracy in complex flows.

Finally, in the context of climate modeling, the most significant issue arises from the introduction of artificial viscosity in many of these models. In order to maintain numerical stability, the viscosity is often increased, even in flows where convection should dominate. Consequently, the flow becomes numerically dominated by diffusion, leading to an undesired smoothing effect (see Eq. 5). This is problematic because flows in climate modeling, such as global wind circulation and ocean currents, are predominantly turbulent and composed of eddies, resulting in a chaotic nature. However, diffusion aims to smooth out the flow, reducing the turbulent eddy-generating behavior. This discrepancy highlights the necessity to explore alternative forms of parameterizations.

Naturally, with the rapid expansion of machine learning and its remarkable accomplishments, researchers have increasingly explored its potential for turbulence parameterization. As a consequence, since 2010, numerous papers have emerged at the intersection of machine learning and fluid dynamics, seeking to leverage these tools.

In 1877, Boussinesq presented his effective turbulent viscosity hypothesis in his work *Théorie analytique de la chaleur*. He proposed that turbulent flows involve momentum exchange between larger, resolved scales, and smaller, unresolved turbulent eddies, which can be represented using an effective viscosity. This relationship links the Reynolds stress to velocity gradients in fluid flow equations (see Eq.14).

Nearly a century later, Pope extended this concept by introducing a more generalized form of the effective-viscosity relationship (Pope, 1975). Instead of assuming a linear relationship, he proposed a power-law relationship. This generalized form allows to capture a broader range of turbulence behaviors and is expected to capture more accurately the dynamics of turbulent flows across various flow regimes.

In 2016, his work resurfaced, emphasizing the invariance property he imposed on the turbulence parameterization he developed. Indeed, in the publication of Ling, Kurzawski, and Templeton, 2016 , they introduced a data-driven, physics-informed parameterization of turbulence, integrating the invariance properties into the neural network architecture. In the context of fluid dynamics, this ensures that the governing equations (see Eq.13, 14 and 15) remain unchanged under specific coordinate transformations and by incorporating them into the DNN design, the model achieves enhanced physical consistency and better captures the physics involved in the turbulent flows. Furthermore, this architecture has served as an inspiration for many others, for an excellent and comprehensive review of them, one should refer to Sharma et al., 2023.

Another great work is the one by Bolton and Zanna, 2019, in their study they use convolutional neural networks to predict unresolved turbulent processes and subsurface flow fields. They demonstrate that these neural networks successfully replicate spatiotemporal variability of subgrid eddy momentum forcing (see Eq.20) and can generalize to various dynamical behaviors while respecting global momentum conservation. This study provides valuable insights for designing ocean eddy parameterizations in coarse-resolution climate models.

As an example of their work (see Fig.3), they investigate the spatiotemporal variability of the true subgrid-forcing term S_x x-components and its predicted value $\tilde{S}_x$ in a 512×512 domain. All the neural networks, denoted by $f_x\left(\bar{\psi}, \mathbf{w}_i\right)$ with $i = 1, 2, 3$, trained on different regions successfully reproduce the spatial patterns of the true S_x but exhibit varying magnitudes. The network trained on data from the western boundary accurately reproduces the correct amplitude and variability. However, the network trained on data from the eastern boundary underestimates the magnitude by approximately 50%, whereas the network trained on data from the southern gyre underestimates S_x by an order of magnitude. To assess their accuracy even more, the Pearson correlation is calculated between S_x and $\tilde{S}_x$. As it can be seen, the networks show high correlation within the jet region but tend toward zero or negative correlation near the eastern boundary.

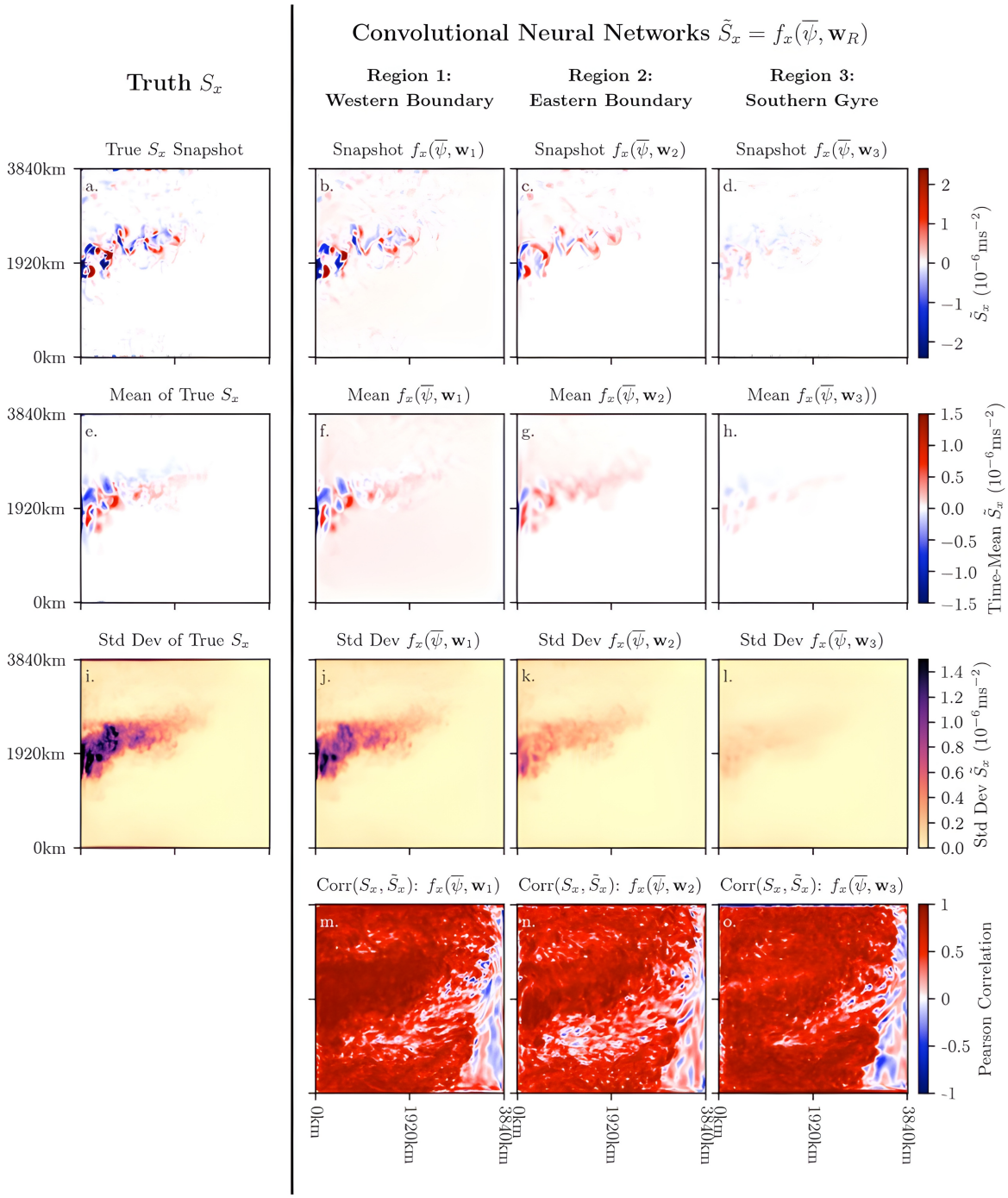


Figure 3: Examining the nonlocal prediction ability. Comparisons of the true zonal component of the subfilter momentum forcing S_x, with the neural networks trained using data from three different regions. The first three rows compare (a-d) snapshots, (e-h) time means, and (i-1) the standard deviation, respectively, while the bottom row (m-o) shows the correlation between the true S_x and the predictions $\tilde{S}_x$. The first column contains the diagnostics using the true zonal subfilter momentum forcing S_x, while columns two, three, and four use predictions $\tilde{S}_x$ from the neural networks $f_x\left(\bar{\psi}, \mathbf{w}_1\right), f_x\left(\bar{\psi}, \mathbf{w}_2\right)$, and $f_x\left(\bar{\psi}, \mathbf{w}_3\right)$, respectively. All diagnostics were produced using the validation data, the figure and legend come from the work Bolton and Zanna, 2019.

Recently, the field of deep learning has witnessed the emergence of an exciting and innovative branch that explores the integration of neural networks within the realm of Fourier space. In fact, this novel architecture, known as the Fourier Neural Operator (Li et al., 2020), represents a significant step forward at the intersection of neural networks and Fourier analysis.

The Fourier Neural Operator (FNO) is innovative because it seamlessly blends the expressive power of neural networks with the unique representation capabilities of Fourier analysis. Neural networks have demonstrated remarkable success in handling complex patterns and capturing intricate relationships within data, while Fourier analysis provides a mathematical framework to decompose signals into frequency components, highlighting long-range interactions and global patterns.

Since then, many new papers building upon it have emerged, including the Factorized Fourier Neural Operator (FFNO), which enhances the architecture by using separable spectral layers, improved residual connections, better training strategies like the Markov assumption, etc (Tran et al., 2021). These advancements allow FFNO to scale to deeper networks and outperform the original Fourier Neural Operator on challenging problems, such as the Navier-Stokes equation. An illustration of its capabilities is shown in Fig.5.

To finish, the Spherical Fourier Neural Operators, recently published, represents a groundbreaking extension of Fourier Neural Operators by leveraging the generalized Fourier transform to efficiently learn operators on spherical geometries (Bonev et al., 2023). Indeed, it provides stable and accurate long-range forecasts for atmospheric dynamics, crucial for climate prediction. An illustration of it is shown in Fig.4. With impressive computational efficiency, SFNOs achieve forecasts for an entire year within 13 minutes on a single GPU. This promising advancement holds significant potential for sub-seasonal-to-seasonal forecasting and machine learning-based climate prediction.

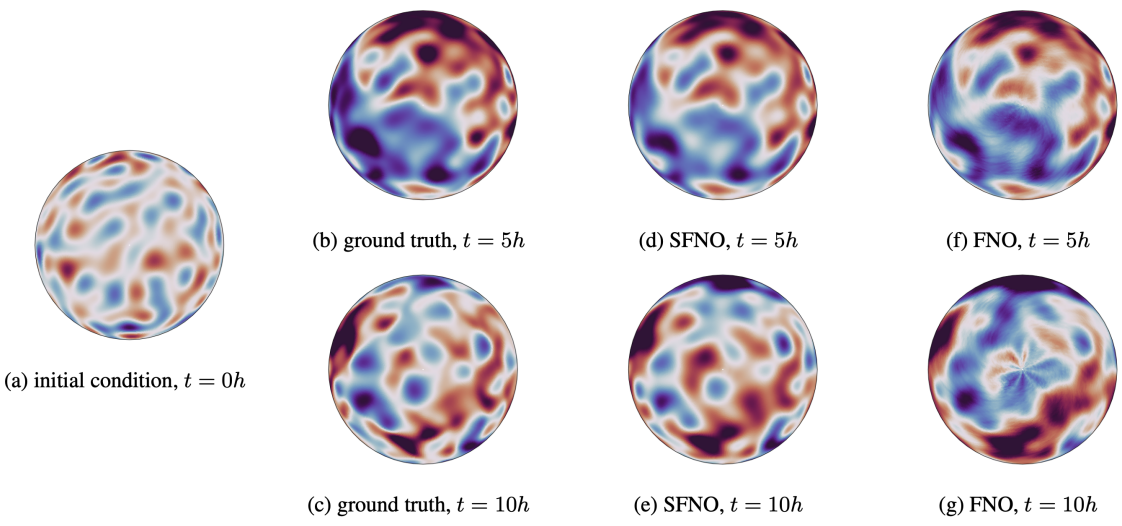


Figure 4: Solutions to the Shallow Water Equations on the rotating Sphere predicted by SFNO and FNO architectures in comparison to the ground truth solution computed using a classical spectral solver. The plots depict the geopotential height at 5 and 10 hours, the figure and legend comes from the work Bonev et al., 2023.

In conclusion, the emergence of these Fourier Neural Operators marks an exciting new frontier in the deep learning field, bringing together the power of neural networks and the insights from Fourier analysis to tackle complex partial differential equation problems with unparalleled efficiency and accuracy. These architectures, the FNO and the FFNO specifficaly, will be explored more deeply in what comes afterwards since they are the object of this thesis.

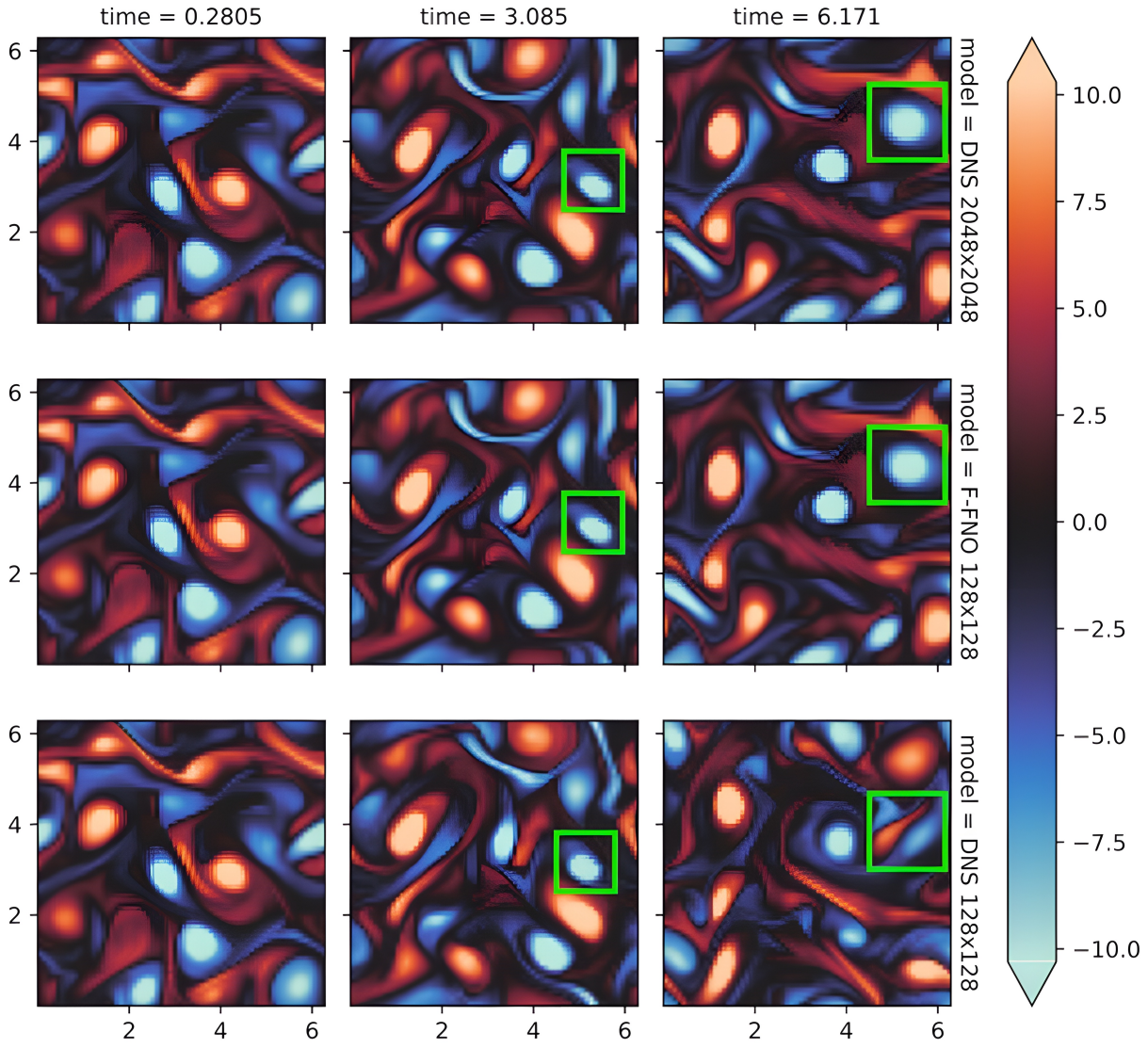


Figure 5: Visualization of the correlation variation between the ground truths and different models. The heatmaps represent the surface of a torus mapped onto a 2D grid, with color representing the vorticity (the spinning motion) of the fluid. One observes that the vorticity fields predicted by the F-FNO trained on 128x128 grids (middle row) correlates with the ground truths (top row) for longer than if the DNS is run using the same spatial resolution (bottom row). This is especially evident after 6 seconds of simulation time (compare the green boxes). In other words, for the same desired accuracy, the F-FNO requires a smaller grid input than a numerical solver, the figure and legend come from the work Tran et al., 2021.

The black-box nature of neural networks, while powerful for approximating complex functions, poses a challenge when it comes to gaining insights into the underlying physical processes. In addition to that, for many complex systems, one only has access to poor models because certain interaction terms are unknown.

A solution to these problems is to use a new method such as the Symbolic Regression (SR) fits, it is a machine learning technique that discovers compact, human-readable mathematical expressions reflecting underlying data relationships, especially useful for complex or unknown variable relationships. SR assumes a sparse and algebraic input-output relationship for the data-generating mechanism. By combining symbolic regression with deep learning, researchers can build hybrid models that retain the flexibility of neural networks while also capturing the interpretability of symbolic equations.

This idea was developped further by Cranmer et al., 2020, they present a novel approach to distill symbolic representations of a deep learning model, focusing on Graph Neural Networks (GNNs). Their technique encourages sparse latent representations during supervised training of GNNs, followed by symbolic regression to extract explicit physical relations from components of the learned model. Moreover, the symbolic expressions extracted using their method generalize better to out-of-distribution data compared to the GNN itself. Applied to a cosmology example, they discover a new analytic formula predicting dark matter concentration from nearby cosmic structures' mass distribution. Hence, applying this method in the context of turbulence closure modeling holds significant promise, as demonstrated by Quattromini et al., 2023, who have successfully utilized Graph Neural Networks for this purpose.

An alternative solution to the initial problem involves using a Relevance Vector Machine (RVM), which differs from symbolic regression methods. The RVM primary objective is to identify the most significant features or variables in the input data that contribute to the prediction task. By estimating the relevance of each feature and automatically selecting the most relevant ones, the RVM creates a sparse model. Zanna and Bolton, 2020 explored this approach and successfully found several parameterizations for the turbulence closure term. Although the results were inferior to those produced by their Convolutional Neural Network (CNN) in Bolton and Zanna, 2019, the RVM provided an interpretable parameterization.

To conclude, regardless of the approach used, whether it involves neural networks, symbolic regression, or other methods, it is essential to assess the quality of the results obtained. A consistent way to achieve this is through the framework presented by Ross et al., 2023. Indeed, this framework enables the evaluation of the offline ability of the parameterization learned, i.e. it uses various metrics to assess the quality of the local predictions made for the closure term. Additionally, the framework facilitates the assessment of the online ability of the network. It provides various tools to observe the quality of simulations conducted at low resolution and corrected at each time step by the network. Key aspects such as the energy spectrum, decorrelation time, probability distribution similarities, and other relevant indicators are also considered. This comprehensive approach ensures a rigorous evaluation of the performance and effectiveness of the parameterizations found.

1.3 RESEARCH OBJECTIVES AND AIMS

The objective of this thesis is to investigate the intersection of deep learning and physics in the domain of ocean dynamics. Specifically, we aim to explore and assess the performance of state-of-the-art deep learning methods in enhancing the quality of low-resolution ocean simulations. The ultimate goal is to discover a new parameterization for the neglected physical processes, i.e. Reynolds stresses (see Eq.20), that could significantly improve the quality of the results, comparable to computationally intensive high-resolution simulations.

To achieve this, the key objectives of this study are as follows:

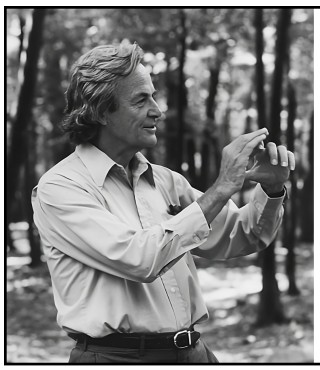
1. Use my background in physics to comprehend the fluid mechanics involved in the case of ocean modelling. This involves explaining and ensuring that everyone possesses sufficient knowledge of the underlying physics to interpret the subsequent results effectively.

2. Reproduce the results obtained in previous studies, namely Bolton and Zanna, 2019; Zanna and Bolton, 2020; Ross et al., 2023, using both their Fully-Convolutional-Neural-Network (FCNN) and their learned parameterization through Relevance Vector Machine (RVM).

3. Extend the investigation conducted in Ross et al., 2023 by employing more complex and larger datasets, with the hope of enhancing the obtained results.

4. Use the Fourier Neural Operators, specifically the FNO (Li et al., 2020) and FFNO (Tran et al., 2021), which is a novel approach in the context subgrid scale processes parameterization, aiming to improve upon the original results and the new results obtained with the expanded datasets.

5. Conduct a rigorous assessment of the newly learned parameterizations, using the benchmark framework established in Ross et al., 2023. This will ensure a comprehensive evaluation of the models performance, accuracy and physical meaning of the resulting simulations.



We absolutely must leave room for doubt or there is no progress and there is no learning. There is no learning without having to pose a question. And a question requires doubt. People search for certainty. But there is no certainty. People are terrified — how can you live and not know?

- Richard P. Feynman

Author : Tamiko Thiel 1984
Contex: Richard Feynman at the Robert Treat Paine Estate in Waltham, Massachusetts, in 1984

Chapter Two

An introduction to fluid dynamics

The objective of this chapter is to introduce the concept of conservation laws in fluid dynamics, derive the Navier-Stokes equations and physically interpret them, introduce the Quasigeostrophy theory, and explore subgrid scale physics parameterizations.

2.1 CONSERVATION LAW

The foundation of fluid dynamics is laid upon conservations laws. The meaning of these laws does not result from mathematics but from physics and observations of the surrounding world. For example, the conservation law for a quantity $\mathbb{U}$ can be stated as:

Definition of a conservation law

The variation of the total amount of a quantity $\mathbb{U}$ inside a given domain is equal to the balance between the amount of that quantity entering and leaving the considered domain, plus the contributions from eventual sources generating that quantity.

Of course, not all flow quantities obey to conservation laws. Indeed, only the mass, momentum and energy do whereas pressure, temperature and entropy do not sastify a conservation law. Now, from a mathematical point of view, the general form of the conservation law can be derived rather easily.

Indeed, in order to understand how $\mathbb{U}$ evolves through time, it is first important to define and understand basic notions such as the one depicted in Fig.6 where:

- Ω is the control volume;
- $\mathbf{F}$ is a flux entering the volume;
- $S = \delta\Omega$ is the boundary of this control volume;
- $d\mathbf{S}$ is the surface element vector pointing along the outward normal;
- Q_V and $\mathbf{Q}_S$ are respectively volume and surface sources with respect to the control volume Ω.

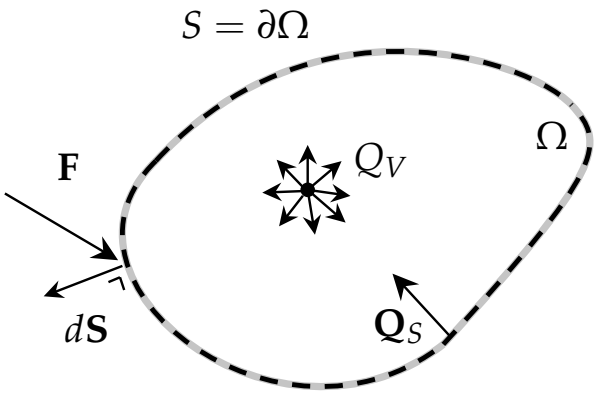


Figure 6: Arbitrary control volume Ω of fluid and representation of basic mathematical notions used in fluid dynamics.

Thus, by translating the conservation law into mathematical terms and using the different notions defined in Fig.6, one obtains:

$$\underbrace{\frac{\partial}{\partial t}\int_{\Omega} \mathbb{U}\, \mathrm{d}\Omega}_{\substack{\text{Change of total} \\ \text{amount of } \mathbb{U} \text{ in } \Omega}} = - \underbrace{\int_{S} \mathbf{F} \cdot \mathrm{d}\mathbf{S}}_{\substack{\text{Amount of } \mathbb{U} \\ \textbf{entering} - \textbf{leaving } \Omega}} + \underbrace{\int_{\Omega} Q_V \, \mathrm{d}\Omega + \int_{S} \mathbf{Q}_S \cdot \mathrm{d}\mathbf{S}}_{\text{Amount of } \mathbb{U} \textbf{ generated}}$$

which corresponds mathematically to the **global form of the conservation law** and it holds true for any abritrary volume Ω. In addition to that, it is important to notice that due to the integrals, it is implied that this law should be verified all over the volume. However, it is also interesting to extract the **local form** of this equation, i.e. the equation that should be satisfied locally over each point contained in the volume. In order to do so, one must use Gauss theorem which is stated as follows:

Gauss theorem

Let Ω be a volume in $\mathbb{R}^3$ and S be the closed surface that encloses Ω. If $\mathbf{F}$ is a flux that is continuously differentiable within Ω and over S, one has:

$$\int_{S} \mathbf{F} \cdot \mathrm{d}\mathbf{S} = \int_{\Omega} \nabla \cdot \mathbf{F} \, \mathrm{d}\Omega \tag{1}$$

where $\nabla \cdot \mathbf{F}$ corresponds to the divergence of the vector field. That is, if $\mathbf{F}$ is a close loop, the divergence would be zero, whereas if the vector field tends to move away from a point, then $\nabla \cdot \mathbf{F}$ would be greater than zero.

In other words, Gauss theorem asserts that the total amount of a given quantity $\mathbb{U}$ only depending on a flux $\mathbf{F}$ in an arbitrary volume Ω can be computed with a volume integral of all sources and sinks or equivalently, using the overall flow passing through the boundary of the volume. An illustration of these integrals is represented in Fig.7.

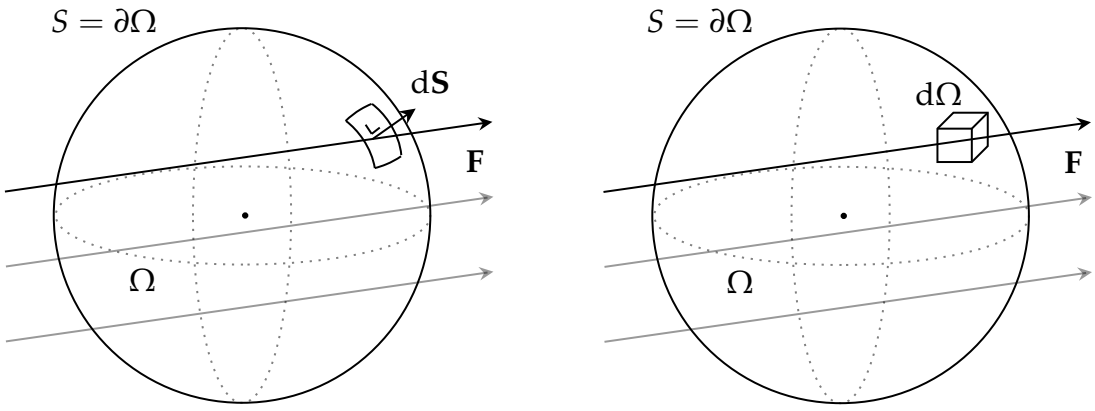


Figure 7: The arbitrary volume Ω contains a quantity $\mathbb{U}$ that depends solely on the flux $\mathbf{F}$. In the left figure, the total amount of $\mathbb{U}$ is computed using the boundary S and the surface element vector $d\mathbf{S}$. In the right figure, the total amount of $\mathbb{U}$ is computed by considering the control volume Ω, the infinitesimal volume $\mathrm{d}\Omega$, and the divergence of $\mathbf{F}$.

Therefore, with the use of Gauss theorem, one obtains:

$$\frac{\partial}{\partial t}\int_\Omega \mathbb{U}\,\mathrm{d}\Omega = -\int_S \mathbf{F}\cdot\mathrm{d}\mathbf{S} + \int_\Omega Q_V\,\mathrm{d}\Omega + \int_S \mathbf{Q}_S\cdot\mathrm{d}\mathbf{S}$$
$$= -\int_\Omega \nabla\cdot\mathbf{F}\,\mathrm{d}\Omega + \int_\Omega Q_V\,\mathrm{d}\Omega + \int_\Omega \nabla\cdot\mathbf{Q}_S\,\mathrm{d}\Omega$$

Since the control volume is fixed, i.e. it does not evolve through time, the time derivative can be moved under the integral sign. In addition to that, this equation holds true for any arbitrary volume Ω hence, both sides are equal with respect to the integrand of the integrals which leads finally to the **local form of the conservation law**:

$$\frac{\partial \mathbb{U}}{\partial t} = -\nabla\cdot\mathbf{F} + Q_V + \nabla\cdot\mathbf{Q}_S$$

In computational fluid dynamics, this equation is extremely important since most of the numerical schemes are based on it and also, it allows to gain insights into the intricate details of fluid behavior at different scales.

Finally, it is important to understand the physical meaning of the flux $\mathbf{F}$ since it plays a major role regarding the behaviour of the flow. In order to do so, one can imagine a fluid flowing through a pipe with a drop of dye placed intially at random as shown in Fig.8. The question is *what influences the evolution of dye concentration through time in the control volume* Ω *?*

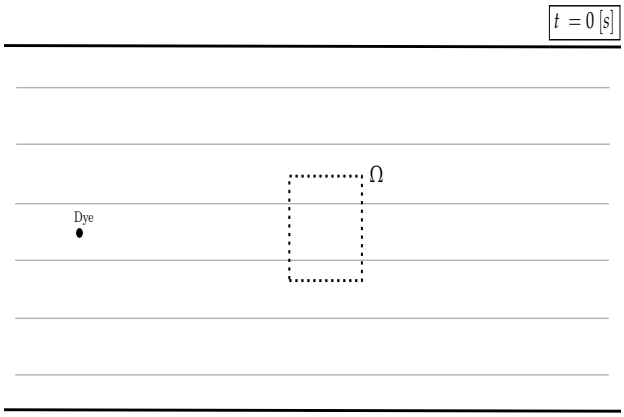
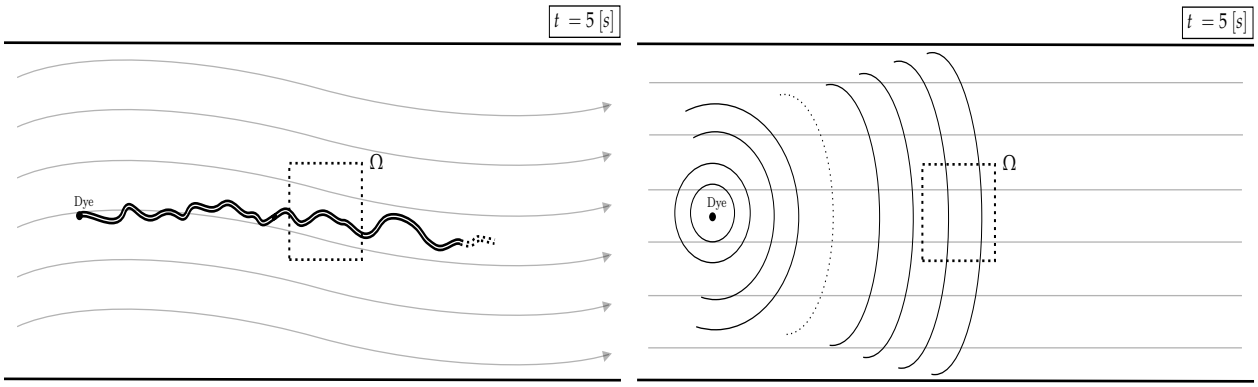


Figure 8: Fluid at rest (straight grey lines) in a pipe (black lines) with some dye (black dot) at time $t = 0$ $[s]$. The control volume Ω defines the zone of interest to observe the evolution of the dye concentration through time.

In a first case scenario, the fluid starts to move forward as shown in Fig.9a and as a consequence, the dye is carried away by the flow. Therefore, the time variation of the dye inside the control volume Ω is induced by a convective flux, i.e. a transport of the dye owing to the fluid flow with a velocity $\mathbf{v}$. The general mahematical expression of this **convective flux** $\mathbf{F}_C$, where U is a given flow quantity and $\mathbf{v}$ the fluid velocity field, is:

$$\mathbf{F}_C = \mathbb{U}\mathbf{v} \tag{2}$$

In the second case scenario represented in Fig.9b, the fluid remains at rest which prevents the transport of dye by the flow. Initially, at the microscopic level, the high concentration of localized dye results in frequent collisions between closely packed molecules due to random thermal molecular movements. These collisions transfer energy, leading to increased movements of neighboring molecules. Consequently, the dye gradually diffuses throughout the system as collisions continue to increase and eventually, the dye diffuses in the control volume Ω. As a result, the time variation of the dye concentration is due to a diffusive flux, i.e. a macroscopic transport driven by microscopic molecular agitation.



(a) (b)

Figure 9: Fluid (grey lines) in a pipe (black line) after some time starting from intial state described by Fig.8. (a) The fluid is moving forward (left to right) and brings along the dye thus leading to a convective flux $\mathbf{F}_C$ in the control volume Ω. (b) The fluid is at rest and the dye diffuses everywhere which leads to a diffusive flux inside the control volume Ω.

From a mathematical point of view, the **diffusive flux** $\mathbf{F}_D$ is expressed as:

$$\mathbf{F}_D = -\kappa \nabla \mathbb{U} \tag{3}$$

where κ is the diffusivity constant and $\mathbb{U}$ a given flow quantity. In conclusion, the key points regarding conservation laws can be summarized as follows:

Global and local conservation law

In a fixed control volume Ω, the **global conservation law** is given by:

$$\frac{\partial}{\partial t} \int_\Omega \mathbb{U} \, \mathrm{d}\Omega = -\int_S \mathbf{F} \cdot \mathrm{d}\mathbf{S} + \int_\Omega Q_V \, \mathrm{d}\Omega + \int_S \mathbf{Q}_S \cdot \mathrm{d}\mathbf{S} \tag{4}$$

Using Gauss theorem (Eq.1), the **local form of the conservation law** is:

$$\frac{\partial \mathbb{U}}{\partial t} + \nabla \cdot (\mathbb{U}\mathbf{v}) = \nabla \cdot (\kappa \nabla \mathbb{U}) + Q_V + \nabla \cdot \mathbf{Q}_S \tag{5}$$

with the flux $\mathbf{F}$ decomposed as the sum of $\mathbf{F}_C$ (Eq.2) and $\mathbf{F}_D$ (Eq.3).

Finally, it is important to emphasize that:

- In a moving fluid, both convection and diffusion processes occur simultaneously, whereas if it is at rest, only diffusion takes place;
- Convection is a non-linear first-order process that enhances momentum in the direction of flow, whereas diffusion is a second-order linear process that disperses momentum in all directions.
- When a fluid is in motion, it can exhibit two distinct patterns: **laminar** flow, characterized by smooth and orderly layers, and **turbulent** flow, featuring chaotic and irregular motion with eddies and swirls. High-speed flows are predominantly influenced by convection, which ultimately leads to a turbulent behavior of the fluid.

2.2 NAVIER-STOKES EQUATIONS

The Navier-Stokes equations constitue a set of partial differential equations used to describe the motion of a fluid, they can be derived from the local conservation law (Eq.5).

2.2.1 MASS CONSERVATION

The law of mass conservation is a general statement that is independant of the nature of the fluid as well as the forces acting on it. As a matter of fact, it expresses that in any fluid system mass cannot disappear from the system, nor be created. Therefore, the flow quantity of interest is simply $\mathbb{U} = \rho$, i.e. the density of the fluid. In addition to that, it is important to notice that mass can only be transported by convection and that no diffusive flux exists thus $\mathbf{F}_C = \rho\mathbf{v}$ and $\mathbf{F}_D = 0$. Finally, in the absence of external sources (Q_V and $\mathbf{Q}_S$ equal to 0), one obtains the **mass conservation law**:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

which in fluid mechanics litterature is often also named the **continuity equation**. It is important to note that, for the sake of simplicity in notations, the spatial ($\mathbf{x}$) and temporal (t) dependencies of all flow quantities have been omitted. In the case where the fluid is **incompressible** meaning that $\rho(\mathbf{x}, t) = \rho$, the density becomes a constant and the mass conservation law can be simplified into:

$$\nabla \cdot \mathbf{v} = 0 \tag{6}$$

This assumption is frequently used in fluid mechanics and is applicable to water at normal speeds, as well as air and liquid gases at low speeds. However, in situations involving high-speed flows, such as those encountered in aeronautics over plane wings, this assumption may no longer hold true.

2.2.2 MOMENTUM CONSERVATION

In physics, momentum is a fundamental concept representing an object resistance to changes in its motion. It is determined by the object mass and velocity, meaning heavier and faster objects possess more momentum, making them more difficult to stop or to alter their path. From a mathematical point of view, momentum is a vector expressed as the product of the mass of an object and its velocity.

For this reason, the quantity of interest is $\mathbb{U} = \rho\mathbf{v}$, i.e. the fluid momentum per unit mass in the 3 space directions. Furthermore, momentum can only be created through convection thus $\mathbf{F}_C = \rho\mathbf{v} \otimes \mathbf{v}$ and $\mathbf{F}_D = 0$. One must notice that momentum has 3 components, thus in this situation $\mathbf{F}_C$ is a **second order tensor** but the notations do not change for simplicity. Moreover, according to Newton's law, t*he variation of momentum is directly proportional to the total force applied to an object*. Actually, this total force can be decomposed as the sum of external and internal forces applied to the fluid. In other words, the source term $\mathbf{Q}_V$ represents the **external volumes forces per unit volume** $\rho\mathbf{f}_E$ whereas $\mathbf{Q}_S$ corresponds to the **sum of internal forces** denoted $\mathbf{f}_i$.

In fluid mechanics, external volumic forces encompass a diverse range of effects, including magnetic forces, Coriolis forces, and others. However, among these forces, gravity stands out as the most familiar. Therefore, as an example, if gravity is the predominant external volumic force, one would obtain the following one-dimensional tensor:

$$\mathbf{Q}_V = \rho \mathbf{f}_E = \rho \mathbf{g} \tag{7}$$

To understand internal forces, one can consider a fluid at rest and zoom in on an infinitesimal volume of fluid denoted $\mathrm{d}\Omega$ as shown in Fig.10. In order to remain in equilibrium the internal stresses experienced by the volume of fluid must be balanced by the external stresses exerted by its surroundings, in accordance with the principle of Newton's action-reaction law.

Thus, there exists an equilibrium of forces in all three-space directions, with each force being the result of contributions from the three directions. As an example, the total force per unit area (= stress) felt by the x-face denoted $\mathbf{f}_x$, can be decomposed as the sum of a stress applied normally to the surface named **pressure** as well as two other tangential stresses to the surface called **shear stresses**. From a mathematical point of view, this can be writen as:

$$\begin{aligned} \mathbf{f}_x &= \sigma_{x,\,x}\ \mathbf{e}_x + \sigma_{y,\,x}\ \mathbf{e}_y + \sigma_{z,\,x}\ \mathbf{e}_z \\ &= \sigma_{i,\,x}\ \mathbf{e}_i \ \ (\textit{Einstein notation}) \end{aligned}$$

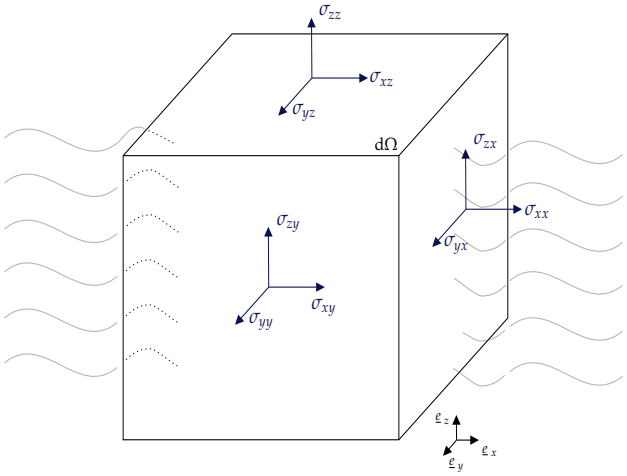


Figure 10: Balance of forces for an infinitesimal volume of fluid $\mathrm{d}\Omega$ at equilibrium with its surrounding in a resting fluid.

where $\sigma_{x,x}$ corresponds to the pressure applied on the x-face, $\sigma_{y,x}$ the shear stress on the x-face in the y-direction, $\sigma_{z,x}$ the shear stress on the x-face in the z-direction and finally $\mathbf{e}_i$ is a unit vector forming the cartesian axis basis with $i = x,\ y,\ z$. Furthermore, the total forces per unit area experienced by the y- and z-faces of the fluid can be derived using a similar reasoning. There are nine stresses in total, typically represented using the internal (or Cauchy) stress second-order tensor where component-wise:

$$\boldsymbol{\sigma} = \begin{cases} \sigma_{\,i,\,i} = p_{\,i,\,i} = \text{Pressure applied on the i-th surface.} \\ \sigma_{\,i,\,j} = \tau_{\,i,\,j} = \text{Shear stress on the j-th surface in the i-th direction.} \end{cases} \tag{8}$$

In conclusion, by combining all the stresses into this second-order tensor, a comprehensive representation of the internal forces is obtained for $\mathbf{Q}_S$ as:

$$\mathbf{Q}_S = \boldsymbol{\sigma} = \mathbf{f}_x\ \mathbf{e}_x + \mathbf{f}_y\ \mathbf{e}_y + \mathbf{f}_y\ \mathbf{e}_z \tag{9}$$

An assumption that is often made in fluid mechanics is that the fluids is Newtonian, i.e. the relationship between the viscosity of the fluid and the shear rate ($\mathrm{d}\tau/\mathrm{d}t$) is linear ! For example, water, air, alcohol are fluids falling into this category. From a mathematical point of view, one can now express the shear stress as:

$$\boldsymbol{\tau} = \mu \left(\left(\nabla \mathbf{v} + \nabla \mathbf{v}^T \right) - \frac{2}{3} (\nabla \cdot \mathbf{v}) \mathbf{I} \right) \tag{10}$$

with μ the dynamic viscosity and $\mathbf{I}$ the identity tensor. Furthermore, under this assumption, the Cauchy stress tensors can be further developped:

$$\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau} = -p\mathbf{I} + \mu \left(\left(\nabla \mathbf{v} + \nabla \mathbf{v}^T \right) - \frac{2}{3} (\nabla \cdot \mathbf{v}) \mathbf{I} \right) \tag{11}$$

Finally, by injecting these former results into the local form of the equation, one will obtain the **momentum conservation law** for Newtonian fluids:

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho (\mathbf{v} \cdot \nabla) \mathbf{v} = -\nabla p + \mu \left(\Delta \mathbf{v} + \frac{1}{3} \nabla (\nabla \cdot \mathbf{v}) \right) + \rho \mathbf{f}_e$$

2.2.3 ENERGY CONSERVATION

In thermodynamic, the energy content of a system can be expressed in terms of its internal energy per unit mass e. Indeed, this internal energy is a state variable hence, its variation during a thermodynamic transformation depends only on the final and initial states. For a fluid, the conserved quantity is the total energy E, which is the sum of its internal energy and kinetic energy per unit mass:

$$E = e + \frac{||\mathbf{v}||^2}{2} \tag{12}$$

In addition to that, it is known that the first law of thermodynamics reveals that the variation in total energy is driven by the work done by the forces acting on the fluid and the heat transferred to it.

Therefore, the quantity of interest is $\mathbb{U} = \rho E$, the convective and diffusive fluxes of energy are respectively given by $\mathbf{F}_C = \rho E \mathbf{v}$ and $\mathbf{F}_D = -\kappa \gamma \rho \nabla e$ since there is no diffusive flux linked to motion with $\gamma = c_p/c_v$, representing the ratio of specific heat coefficients under constant pressure and constant volume. Additionally, the external volume source Q_V corresponds to the sum of the work done by the volume forces $\mathbf{f}_e$ and the heat sources q_H, such as radiation, heat released by chemical reactions, etc. Finally, the internal source $\mathbf{Q}_S$ corresponds to the work done by the internal shear stresses acting at the surface of the fluid mathematically expressed as $\boldsymbol{\sigma} \cdot \mathbf{v}$. As a result, the **energy conservation law** is:

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot (\rho \mathbf{v} E) = \nabla \cdot (k \nabla T) + \nabla \cdot [(-p\mathbf{I} + \tau) \cdot \mathbf{v}] + \rho \mathbf{f}_\mathrm{e} \cdot \mathbf{v} + q_H$$

In conclusion, the Navier-Stokes equations can be summarized as follows:

Navier-Stokes equations

For a Newtonian fluid (Eq.10), the **mass conservation law** reads as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{13}$$

In addition to that, the **momentum conservation law** is expressed as:

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho(\mathbf{v} \cdot \nabla)\mathbf{v} = -\nabla p + \mu \left(\Delta \mathbf{v} + \frac{1}{3} \nabla (\nabla \cdot \mathbf{v}) \right) + \rho \mathbf{f}_e \tag{14}$$

Finally, the **energy conservation law** is given by:

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot (\rho \mathbf{v} E) = \nabla \cdot (\kappa \gamma \rho \nabla e) + \nabla \cdot (\boldsymbol{\sigma} \cdot \mathbf{v}) + \rho \mathbf{f}_\mathrm{e} \cdot \mathbf{v} + q_H \tag{15}$$

2.3 QUASIGEOSTROPHY

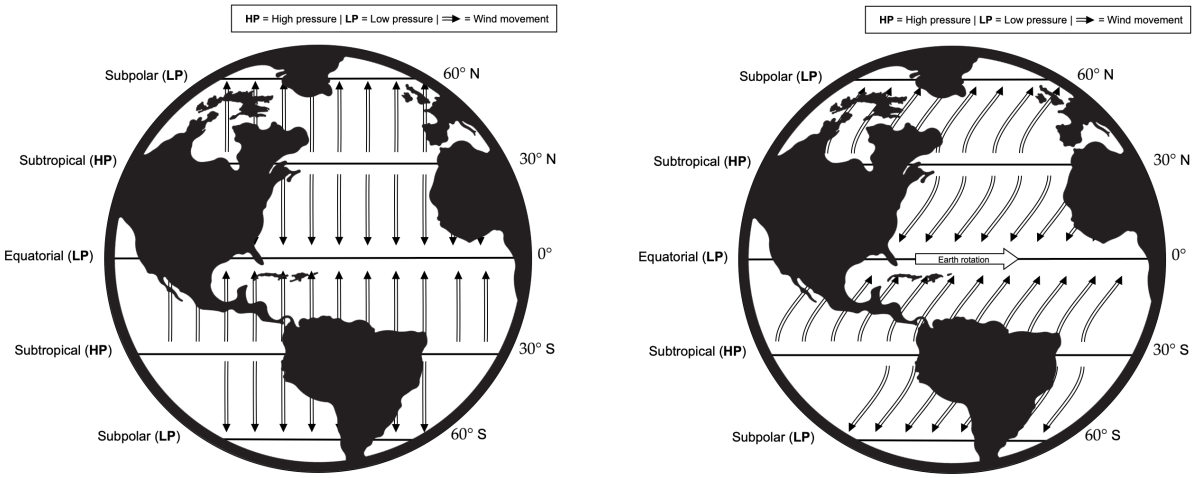
The Navier-Stokes equations are non-linear partial differential equations that aim, once solved, to completely describe the motion of a fluid. However, solving these equations directly without any simplification is extremely computationally expensive. Therefore, it is essential to understand the physics involved and needed for simulating the oceans to simplify the equations using certain assumptions. Consequently, one must first understand the main causes of wind at the ocean and atmospheric scale.

2.3.1 PRESSURE GRADIENT

The air applies pressure over Earth's surface due to its weight. The air densitiy varies inversely with the temperature; its density decreases when the temperature increases because molecules have more energy and are able to move further away from each other. In addition to that, due to the slight inclination of its axis of rotation, Earth is not uniformly heated by the Sun, which creates a temperature gradient all over its surface. Therefore, high and low-pressure regions are created, leading to the creation of wind, i.e. air is pushed from high to low-pressure regions. An illustration of these pressure zones as well as the idealized wind movement (without taking Coriolis force into account) are represented in Fig.11(a).

2.3.2 CORIOLIS FORCE

The Coriolis force is an apparent force resulting from Earth's rotation, leading to the deflection of moving objects within a rotating reference frame. In the Northern Hemisphere, objects veer to the right, while in the Southern Hemisphere, they veer to the left. The strength of this force is influenced by an object's speed, direction of motion, and latitude. Moving closer to the poles brings one nearer to the axis of rotation, resulting in an increased rotational speed. Consequently, the Coriolis force becomes more pronounced in these regions due to the higher rotational speed. An illustration of the global wind circulation on Earth which takes into account pressure gradients as well as Coriolis force is shown in Fig.11(b).



(a) (b)

Figure 11: Illustration of Earth's wind circulation. (a) In this situation, only pressure gradients created by the high and low-pressure regions are taken into consideration, leading to straight wind directions. (b) In this second situation, Earth's rotation is also considered through the Coriolis force, which curves wind trajectories.

In situations where the dominant physical causes influencing the flow are the pressure gradient and the Coriolis force, and they precisely balance out, a **Geostrophic flow** emerges. A jet stream is a good example of a geostrophic flow, it is characterized by high velocity and narrow air currents. Typically, these flow patterns are commonly observed from the upper part of the mid-latitude ($\sim 45°$) to the pole, where the Coriolis force is at its strongest. However, in regions where the Coriolis force weakens, such as in the subtropic regions (around $\sim 30°$ latitude), convection becomes more influential. As a result, the initial jet-like structure slowly evolves into a more chaotic flow. In other words, the straight and narrow path taken by the flow widens, and eddies begin to form. The flow described here is known as a **Quasi-geostrophic** flow.

The quasigeostrophy theory proves to be useful in atmospheric and oceanic fluid dynamics since it simplifies the Navier-Stokes by neglecting certain terms. This simplification makes the models computationally efficient while effectively capturing the essential features of geophysical phenomena, including jet streams, ocean currents, mesoscale eddies, and weather systems.

As it happens, mesoscale eddies (around 10 to 100 kilometers) play essential roles in various geophysical phenomena. In the atmosphere, they contribute to the development and evolution of weather systems, such as cyclones and anticyclones. Similarly, in the ocean, they transport heat, nutrients, and momentum, influencing the distribution of temperature, salinity, and currents. Additionally, the interactions of eddies with larger-scale flows lead to the transfer of energy and the redistribution of properties within the fluid. Consequently, the importance of the Quasigeostrophy becomes evident, as it allows for modeling these eddies, and understanding their dynamics is crucial for improving weather forecasts, climate models, primary production cycles and our overall comprehension of Earth's atmospheric and oceanic systems.

2.3.3 PYTHON QUASIGEOSTROPHIC MODEL (*PYQG*)

The numerical simulations are done with *PyQG*, it is a python library that models quasi-geostrophic systems using pseudo-spectral methods. The derivation of the complete set of quasigeostrophic equations is left to the reader, Griffies, 2018 is a great book to help doing it. Therefore, one will only be introduced to the prognostic variable, i.e. the physical quantity that allows to determine the state of the system.

First of all, the model used is a **two-layer quasigeostrophic flow** which simplifies the governing equations of fluid dynamics (Eq.13, 14 and 15) while preserving important geophysical flow features. Indeed, it reduces computational complexity and approximates vertical structure by dividing the fluid into two layers with varying densities. This allows to capture essential dynamics like baroclinic instability and vertical motion. A representation of the model is shown in Fig.12.

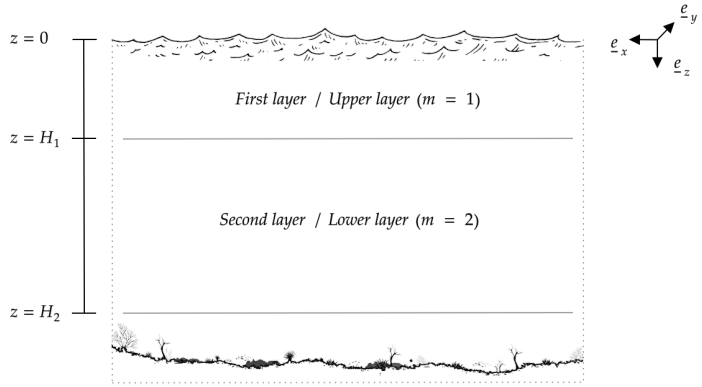


Figure 12: Illustration of the two-layer quasigeostrophic flow model solved using PyQG to generate data. This is a side view showing the different layers but the model is 2-dimensional, i.e. the results obtained are in the XY-plane.

The model prognostic variable is the **potential vorticity** q, it is a conserved quantity in an inviscid, adiabatic, frictionless fluid and is crucial to understand large-scale atmospheric and oceanic flows. As an example, illustrated in Fig.13, one imagine observing a moving air column at a fixed latitude (Coriolis force is constant). It's mathematical expression is given by:

$$q = \frac{\zeta + f}{\Delta z} = \mathrm{C}^{\mathrm{st}} \qquad (16)$$

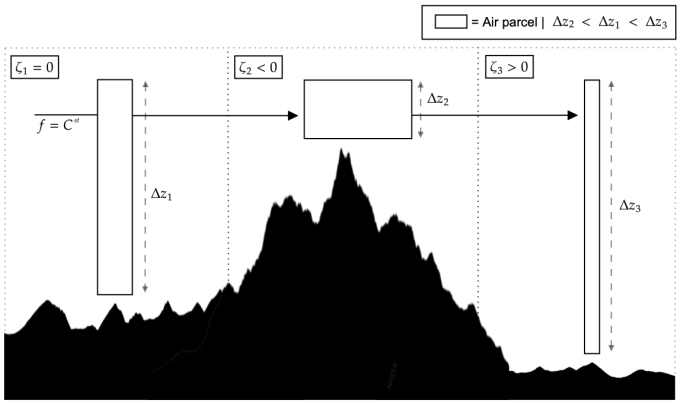


Figure 13: Evolution of the relative vorticity ζ of a moving fluid parcel observed at constant lattitude ($f = \mathrm{C}^{\mathrm{st}}$) for 3 regions of different heights Δz.

with ζ the relative vorticity, f the planetary vorticity and Δz the height of the air column. Initially $\zeta_1 = 0$, therefore by conservation, one must have $\zeta_2 < 0$ in the second region which implies that the air column turns clockwise (if in the Nothern hemisphere) and it is unstable. Finally, in the third region, one obtains $\zeta_3 > 0$ leading to a counter-clockwise movement and a more stable flow. Hence, one understands easily the usefullness of q to determine the dynamics of a flow, since it is a conserved quantity, it takes into account the vertical motion and it is a scalar thus, it does not depend on the coordinates system used.

The **potential vorticity of a two-layer quasigeostrophic flow** is:

$$q_m = \nabla^2 \psi_m + (-1)^m \frac{f_0^2}{g' H_m} \Delta\psi \quad \text{with } m \in \{1, 2\} \tag{17}$$

where ψ_m denotes the streamfunction at depth H_m and the difference between the two layers is given by $\Delta\psi = \psi_1 - \psi_2$. The operator ∇ represents the horizontal gradient, and g' stands for the reduced gravity. Additionally, the planetary vorticity f is calculated using a beta plane approximation, which assumes linearity with respect to latitude. Thus, $f = f_0 + \beta y$, where f_0 represents the Coriolis parameter and β corresponds to the slope. In addition to that, the velocity vector for the m-layer is expressed as $\mathbf{v}_m =< u_m, v_m >$, where u_m and v_m correspond to the longitudinal and latitudinal velocities respectively. The **prognostic equation** in layer m, solved using a pseudo-spectral, is given by:

$$\frac{\partial q_m}{\partial t} + (\mathbf{v}_m \cdot \nabla) q_m = -\beta_m \frac{\partial \psi_m}{\partial x} - U_m \frac{\partial q_m}{\partial x} - \delta_{m,2}\, r_{ek} \nabla^2 \psi_2 + \text{ssd} \tag{18}$$

where $\beta_m = \beta + (-1)^{m+1} f_0^2 / (g' H_m)$ represents the mean potential vorticity gradient, $\Delta U = U_1 - U_2$ is a constant mean zonal velocity shear between the two-fluid layers and *ssd* stands for small scale dissipation. Additionally, the Dirac delta function $\delta_{m,2}$ indicates that the bottom drag with coefficient r_{ek} is only applied to the second layer due to interaction with the ocean floor. Finally, in the spectral space denoted by the symbol $\widehat{()}$, the streamfunctions can be obtained from the PV as follows:

$$\left(\mathbf{M} - \kappa^2 \mathbf{I}\right) \cdot \begin{bmatrix} \hat{\psi}_1 \\ \hat{\psi}_2 \end{bmatrix} = \begin{bmatrix} \hat{q}_1 \\ \hat{q}_2 \end{bmatrix}, \text{ where } \mathbf{M} = \begin{bmatrix} -\frac{f_0^2}{g' H_1} & \frac{f_0^2}{g' H_1} \\ \frac{f_0^2}{g' H_2} & -\frac{f_0^2}{g' H_2} \end{bmatrix} \tag{19}$$

Here, $\kappa = \sqrt{k^2 + l^2}$ represents the radial wavenumber, where k and l are the zonal and meridional wavenumbers, respectively. Finally, the complete dynamic of the fluid can be determined, as the velocity fields can be obtained from the streamfunctions using the relationships $u_m = -\partial_y \phi_m$ and $v_m = \partial_x \phi_m$.

2.4 SUBGRID PHYSICS AND PARAMETERIZATIONS

To solve the Eq.18, it is necessary to discretize both the equation and the spatial domain. If solved anatically, the solution describes the behaviour of the potential vorticity at any spatial scale. However, due to the discretization of the domain, only the physical phenomena occuring at a size greater than the numerical resolution are captured while smaller scale physics is unsolved. For that reason, a cell value can be seen as the average value of all the contributions coming from physical phenomena occuring at a smaller scale inside of it.

Hence, with increasing simulation resolution, the level of detail in the physics improves, and errors due to neglected physical processes decrease. However, higher resolution simulations come with a significant computational cost, especially in the case of earth climate simulations. To address this challenge, one solution is to conduct low-resolution simulations and use a **parametrization** to account for the missing contributions. To this day, the development of these parametrizations is an active and crucial area of research at the intersection of turbulent fluid mechanics and machine learning.

In order to understand what needs to be parameterized, one must first assume that the exact value of a given flow quantity $\mathbb{U}$ can be decomposed as:

$$\underbrace{\tilde{\mathbb{U}}}_{\text{Total}} = \underbrace{\bar{\mathbb{U}}}_{\text{Mean}} + \underbrace{\mathbb{U}'}_{\text{Fluctuation}}$$

Numerically speaking, the cell value corresponds to the average solution $\bar{\mathbb{U}}$ and the missing contribution of the negelected physical processes is represented by $\mathbb{U}'$. Therefore, what is truly solved numerically is the average prognostic equation. As an example, one can imagine solving the momentum conservation law given by Eq.14 but for simplicity the linear terms are grouped into **F** (forcing terms) and **D** (dissipation terms). Hence, averaging the equation is done as follows:

$$\overline{\frac{\partial \mathbf{v}}{\partial t}} + \overline{(\mathbf{v}\cdot\nabla)\mathbf{v}} = \overline{\mathbf{F}} + \overline{\mathbf{D}} \quad \Longleftrightarrow \quad \frac{\partial \overline{\mathbf{v}}}{\partial t} + \overline{(\mathbf{v}\cdot\nabla)\mathbf{v}} = \overline{\mathbf{F}} + \overline{\mathbf{D}}$$

As it can be seen, the non-linear advection term needs more work to be developped properly. For this reason, assuming that the mean value of the fluctuation is equal to zero and using *Einstein notation*, one obtains:

$$\begin{aligned}
\overline{(\mathbf{v}\cdot\nabla)\mathbf{v}} = \overline{\tilde{v}_j \frac{\partial \tilde{v}_i}{\partial x_j}} &= \overline{\left(\bar{v}_j + v'_j\right)\frac{\partial}{\partial x_j}\left(\bar{v}_i + v'_i\right)} \\
&= \overline{\bar{v}_j \frac{\partial \bar{v}_i}{\partial x_j}} + \overline{\bar{v}_j \frac{\partial v'_i}{\partial x_j}} + \overline{v'_j \frac{\partial \bar{v}_i}{\partial x_j}} + \overline{v'_j \frac{\partial v'_i}{\partial x_j}} \quad \text{where} \quad \overline{\bar{v}_j \frac{\partial v'_i}{\partial x_j}} = \overline{v'_j \frac{\partial \bar{v}_i}{\partial x_j}} = 0 \\
&= \bar{v}_j \frac{\partial \bar{v}_i}{\partial x_j} + \frac{\partial}{\partial x_j}\overline{v'_i v'_j} \\
&= (\overline{\mathbf{v}}\cdot\nabla)\overline{\mathbf{v}} + \frac{\partial}{\partial x_j}\overline{v'_i v'_j}
\end{aligned}$$

In conclusion, the average momentum equation solved numerically is expressed as:

$$\frac{\partial \overline{\mathbf{v}}}{\partial t} + (\overline{\mathbf{v}}\cdot\nabla)\overline{\mathbf{v}} = \overline{\mathbf{F}} + \overline{\mathbf{D}} + \mathbf{S} \tag{20}$$

where **S** incorporates contributions arising from physics occurring at a smaller scale than the resolution. In fluid mechanics literature, this term is referred to as the **subfiltered momentum**. It mathematically corresponds to the divergence of the mean fluctuations product, but from a physical standpoint, it lacks a clear interpretation. Consequently, deriving an intuitive analytical expression for this term has been a longstanding challenge within the scientific community.

It is also important to note that the mean fluctuations product, i.e. $\overline{v'_i v'_j}$, is known as the **Reynolds stresses** in the literature. Additionally, Eq.22 is not the only representation of the subgrid term. In fact, there are two other possibilities. The first one is obtained by formulating the problem such that the numerical model requires a parameterization of the Reynolds stresses. Another option involves defining subgrid terms based on the difference between the prognostic equations solved at high and low resolution.

In this example, the expression for the subfilter momentum has been found. With a similar and more complex reasoning, one can extract the same expressions for the potential vorticity, which will be used throughout this work.

Subgrid tendency formulations for potential vorticity

First, denoting ∂_t^H and ∂_t^L as the tendency equations (see Eq.18) from the high- and low-resolution models respectively, the **total subgrid forcing** is given by:

$$S_{q_{tot}} = \overline{\partial_t^H q} - \partial_t^L \overline{q} \tag{21}$$

Alternatively, one can consider the **subgrid forcing of potential vorticity** resulting from unresolved non-linear advection:

$$S_q = \overline{(\mathbf{v} \cdot \nabla) q} - (\overline{\mathbf{v}} \cdot \nabla)\overline{q} \tag{22}$$

Lastly, the subgrid flux, i.e., Reynolds stresses, can be considered. By finding a parameterization of it and applying a numerical divergence operation, it ensures that the added quantity results from the divergence of some quantity, thus respecting the conservation law from a mathematical standpoint. The mathematical expression of the **subgrid flux** is:

$$\mathbf{\Phi}_q = \overline{\mathbf{u} q} - \overline{\mathbf{u}}\,\overline{q} \tag{23}$$

Under the assumption of an incompressible flow (see Eq.6) and that differentiation commutes with filtering and coarsening, one finds that $\overline{\nabla} \cdot \phi_q \approx S_q$ (Ross et al., 2023). In practice, these three formulations are highly correlated and nearly identical.

As a conclusion to this chapter, Fig.14 is a final illustration of the impact of the model resolution on the dynamic of the solution obtained.

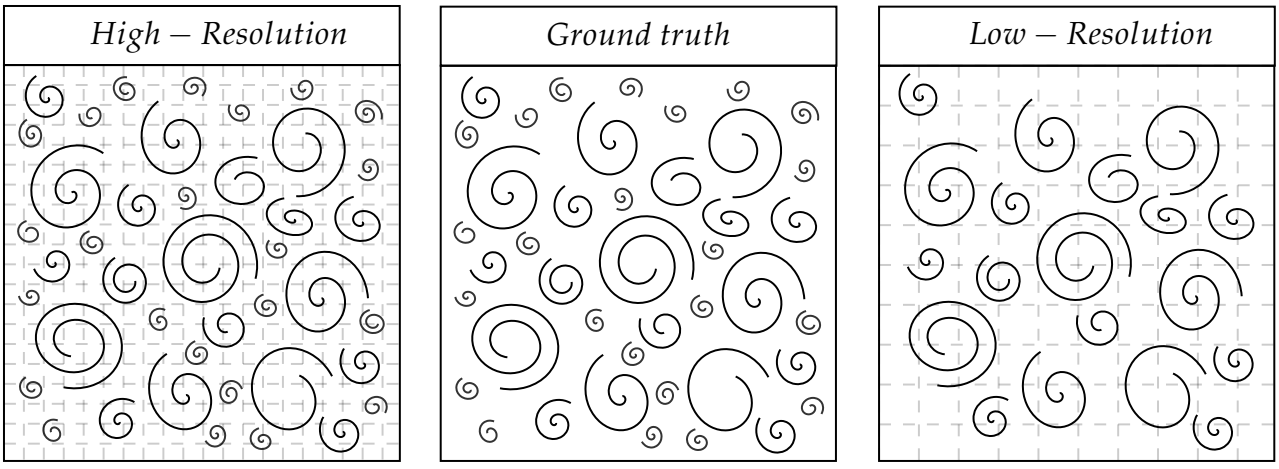


Figure 14: Impact of numerical resolution on the dynamic of the solution. In the high-resolution simulation, small-scale eddies have dimensions higher than the numerical resolution, while in the low-resolution simulation, small eddies remain unresolved.

Chapter Three

Neural Networks and Neural Operators

This chapter aims to introduce the neural networks used throughout this study. These networks fall into two main groups: those performing computations in the spatiotemporal domain and those operating in the frequency domain. Networks from the first group will be briefly introduced, while those from the second group will receive more detailed explanation due to their innovative nature.

3.1 FULLY CONVOLUTIONAL NEURAL NETWORK

First of all, the initial neural network used is a fully-convolutional neural network (FCNN), designed for image analysis tasks. Indeed, FCNNs are built using convolutional layers that excel at recognizing patterns in visual data like images. Unlike typical convolutional neural networks that end with fully connected layers for classification, they maintain the convolutional structure throughout the network.

Convolutional layers in FCNNs retain image details, making them valuable for tasks demanding accurate pixel-level outcomes. As a matter of fact, they are beneficial for tasks like identifying object edges and in domains such as ocean dynamics simulations. In these simulations, FCNNs can predict subgrid process contributions at the pixel level using input snapshots of flow quantities (Bolton and Zanna, 2019). The architecture of the FCNN, capabilities, and constraints have been examined deeply in Ross et al., 2023. However, opportunities for enhancement remain, as they have not explored the impact of more complex training datasets, which was a matter left for future investigation. Therefore, the first objective consist to improve upon the results presented in their research.

3.2 U-NET

The **U-Net**, initially introduced by Ronneberger, Fischer, and Brox, 2015, is a notable neural network architecture widely used for image segmentation tasks. Indeed, its distinctive U-shaped design is effective in capturing complex spatial patterns in images. More precisely, the architecture includes a contracting path to understand contextual information, followed by an expanding path for accurate object or region localization. The U-Net has gained a reputation as a leading deep learning architecture (Çiçek et al., 2016). Thus, the second objective is to assess the U-Net capabilities compared to the FCNN. This completes our set of two neural networks known for excellence across various applications, while operating in the spatial and temporal domains.

Both neural networks are illustrated in Fig.15 to easily observe and compare their architectures. In addition to that, their configuration used are given in the Tab.

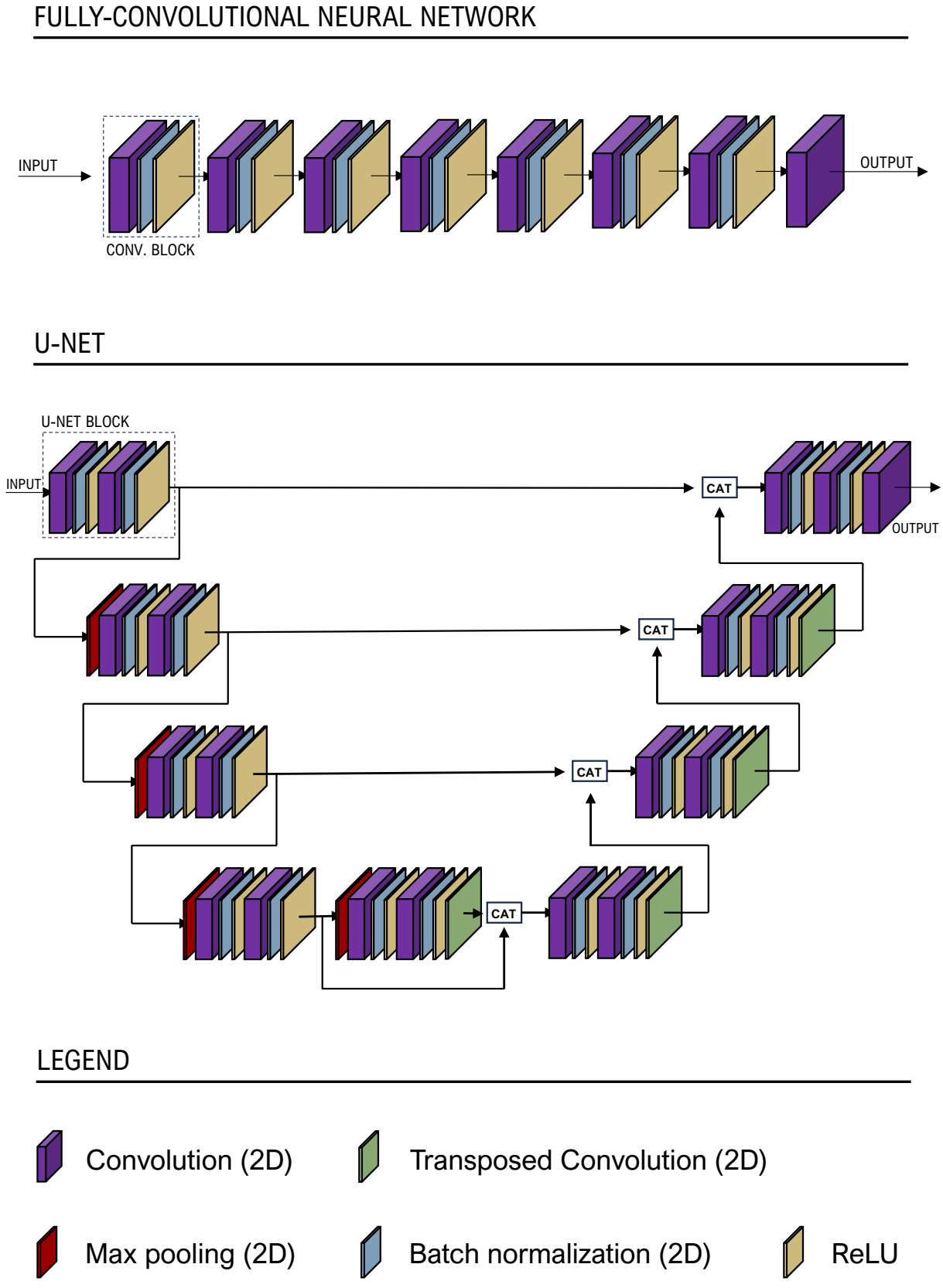


Figure 15: Vizualisation of the fully-convolutional neural network (FCNN) and U-NET architectures. In the figure, **CAT** means the concatenation of the inputs along the batch dimension.

3.3 FOURIER NEURAL OPERATORS

The Fourier Neural Operators (FNO) provide an innovative approach to solving complex partial differential equations, perfect examples of equations are Eq.13, 14 and 15, by combining the capabilities of neural networks with Fourier analysis Li et al., 2020. These operators address the limitations of conventional solvers and introduce a way to model mappings between infinite-dimensional function spaces.

First of all, **Fourier analysis** is a mathematical technique used to understand and break down complex patterns or signals into simpler components. It reveals the underlying structure of a signal by showing how much different frequencies contribute to it. As a puzzle that can be assembled from smaller pieces, any complex signal can be thought of as being made up of different piece with various frequencies.

Hence, Fourier analysis allows to identify these pieces and their strengths, helping make sense of the original pattern. This technique has applications in various fields, from understanding sound and images to solving scientific and engineering problems that involve waves and oscillations. An illustration of the application of Fourier analysis to a complex signal is depicted in Fig.16.

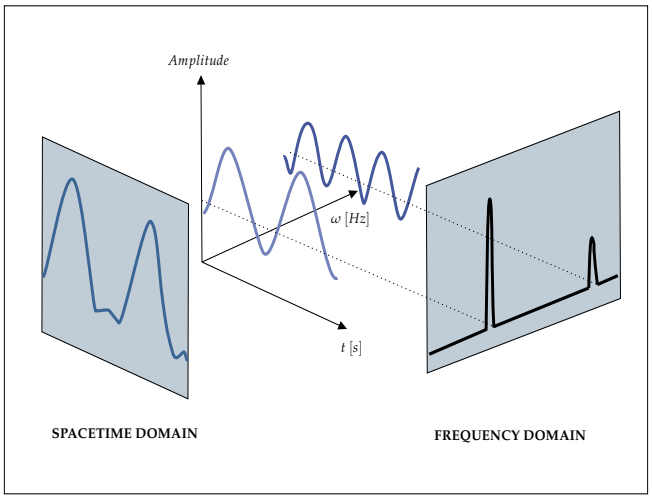


Figure 16: Illustration of the link between spacetime and frequency domains. On the right, the input signal, with varying amplitude over time, can be broken down into the sum of two basic signals (middle). In the frequency domain, its representation consists of two peaks whose amplitude corresponds to the amplitude of the basic signals. These peaks are observed at a specific oscillation frequency ω.

As mentioned earlier, the main idea behind the Fourier neural Operator is to establish a mapping between infinite-dimensional function spaces. To illustrate this somewhat complex notion, one can imagine addressing a challenge like predicting temperature distribution on a wall surface based on position coordinates (x, y). One solution could be to use a conventional neural network to learn this relationship, connecting input coordinates to output temperatures. Nevertheless, this approach proves to be effective only for specific instances of the problem, i.e. if the data was generated using constant conditions like uniform thermal conductivity or a purely vertical temperature gradient.

The Fourier neural operator introduces a more profound perspective. Instead of focusing solely on predicting outputs from inputs, it considers the broader context. As a matter of fact, it acknowledges that there exists a function to describe the initial state of the system and another function to represent how the system evolves over time. Both of these functions are defined over infinite-dimensional spaces meaning that, there exist an infinite number of ways of formulating this function that would give the same mapping.

As a matter of fact, the FNO aims to learn the very equation that governs the system behavior. This marks a departure from traditional neural networks, which often specialize in particular scenarios. With the Fourier neural operator, the underlying physics or equations become the target of learning. In other words, this means that even if the conditions change, such as altering thermal conductivity or other physical parameters, the neural network has the potential to adapt because it is effectively learning the fundamental relationships.For fluid dynamics, it will focus on learning the governing simplified Navier-Stokes equation regardless of the parameters used to impose the physics of the flow.

3.3.1 THE ARCHITECTURE

The flow of input through the Fourier Neural Operators can be described as follows:

- **Domain transformation**: Instead of discrete indices (i, j), the FNO uses real coordinates (x, y) for continuous domain representation. Real coordinates align naturally with real-world situations and provide intuitive interpretations for variables like positions, temperatures, etc. . Unlike indices tied to specific grid sizes, real coordinates offer flexibility for various resolutions, crucial for accuracy and efficiency. They also encourage generalization across scenarios, enabling architecture and weight reuse. This adaptability enhances the neural operator efficiency and effectiveness.

- **Projection** : In the second step, the input, now represented as the continuous function $a(x,\ y)$ in $\mathbb{R}^{X\times Y\times 1}$, undergoes projection into a higher-dimensional space. Here, x and y exist in $\mathbb{R}$, with X as the number of grid points along the x-direction, and Y for the y-direction. Indeed, this projection takes the input from the two-dimensional plane to a space with more dimensions which enriches the original input, capturing more intricate relationships.

  In particular, this increased complexity in the higher-dimensional space enhances the neural operators understanding of the input patterns and features. In simpler terms, the projection extends the input information by examining it from various perspectives in this expanded space which allows the neural operators to extract more detailed information. From now on, the input is mathematically written as $v(x,y) \in \mathbb{R}^{X\times Y\times H}$ with H the size of this latent space on which the initial input is projected.

  It is important highlighting that the **projection is performed pixel-wise** using a simple single-layer feedforward network. This approach treats each input point independently, making the projection unaffected by the grid size which ensures the neural operators effectiveness across various input resolutions. Furthermore, by handling pixels individually, the neural network can learn versatile patterns not tied to a single problem. This enables the network to extract useful features applicable to a range of problems, enhancing its adaptability to new tasks.

- **Fourier layer** : The computations performed can be separated in two different computational path:

    1. **Frequency Domain**: In this first step, the input signal $v(x,\ y)$ undergoes a 2D Fourier transform, this process breaks down and extracts the frequency components of the original input. These components, often referred to as *modes*, make up the decomposed signal. Each of these modes has contributions in both the x and y directions and the total number of modes, in a specific direction, equals the resolution of the image in the corresponding direction. As explained previously, the signal is represented as a sum of terms but it is important to noticte that the initial term in the series holds the strongest influence over the signal. Indeed, as one moves deeper into the terms, their impact diminishes. In other words, the first Fourier modes represent signals with larger amplitudes and lower frequencies. In the context of fluid simulations, this low-frequency content in spacetime corresponds to the contribution of large eddies in turbulent flows to the overall dynamics. Whereas, higher mode values are linked to high-frequency content, representing smaller eddies with rapid motion and evolution.

       After applying the 2D Fourier transform, a low-pass filter is used on the frequency content, selecting only certain modes from the lowest range. The primary goal is to train the neural network to capture information from the global input pattern. As an example, the neural operator can learn from 64x64 resolution fluid flow snapshots to predict energy corrections while focusing on the initial 8 modes out of the available 64. This approach ensures versatility: when correcting a 32x32 simulation, the neural operator can generalize because the first 8 modes are available to both resolutions. Another example, if time limits allow creating only a vast 64x64 dataset for training, the neural operator remains afterwards useful for refining higher-resolution simulations, like 128x128. Once again, this is possible due to its training in making corrections while considering the broader flow pattern.

       Now, one can finally witness core concept of the Fourier neural operator, the retained modes of the original input are convolved to derive the output frequency content. The clear advantage lies in conducting this convolution in the frequency domain. Indeed, from a mathematical point of view, performing convolution between an input and a given kernel is computationally intensive in the space-time domain, while in the frequency domain, it simplifies to a product between the two. The kernel, represented as a matrix of weights denoted as $R \in \mathbb{C}^{H\times H\times M}$ with M the number of modes kept, responsible for multiplying the residual frequency spectrum of the original input, is learned by the neural operator during training. Each Fourier layer possesses its individual weight set and from it the output spectrum is derived, an inverse Fourier transform is used to obtain the output space-time representation.

    2. **Space time domain** : In this second step, the input signal $v(x,y)$ undergoes a simple linear transformation using a pixel-wise 2D convolution.

Lastly, the transformed inputs are combined, and if the Fourier layer is not the final one, the result is subjected to a *GeLU* activation function.

- **Back projection**: At the last Fourier layer, each pixel is individually transformed to the intended output dimension through two separate single-layer feedforward networks. The first network is used to facilitate the transition between the latent space and an intermediate space of fixed size 128, followed by the second network that creates the transition from this space to the final target dimension. Ultimately, the final output is mathematically represented as $u(x,\ y) \in \mathbb{R}^{X \times Y \times O}$, where O denotes the target output dimension.

3.4 FACTORIZED FOURIER NEURAL OPERATORS

The Factorized Fourier Neural Operator (FFNO), as introduced by Tran et al., 2021, presents an enhanced architecture derived from the original Fourier Neural Operator. Notably, the key improvements comes from modifications within the Fourier Layer:

- The 2D Fourier transform is applied to extract individually the modes along the x-direction and y-direction. Subsequently, each set of modes is convolved with its dedicated weight matrix (R_x and R_y), both of which the Neural Operator has to learn.
- Eventually, the outputs, corresponding respectively to distinct part of the truncated frequency spectrum (as in FNO), are transformed back to the spacetime domain and then added to one another.
- The result then flows through a feedforward network composed of a single-layer feedforward network followed by a dropout, *ReLU* activation (unless it's the final Fourier layer) and a layer normalisation.
- The original input is added to the feedforward network output to act as a skip connection, this combined result exits the Fourier Layer.

Alongside this updated Fourier layer design, the inclusion of shared weight matrices R_x and R_y was introduced. This choice slightly impacts model accuracy while significantly reducing the overall number of trainable parameters in the final Neural Operator. Also, the original FNO faced a scaling problem: using more than 4 Fourier layers led to poor training and results. The adjustment made regarding the residual connections enabled the Neural Network to scale effectively, accommodating up to 24 Fourier layers. The Fourier Neural Operator and the Factorized Fourier Neural Operator, like the initial two neural networks, are depicted in Figure X.

Note: The FFNO architecture will be studied in more depth compared to the others, which requires memory-intensive training. As a result, training the neural network needed multiple GPUs. However, the architecture code was incompatible with parallel execution due to its use of (torch) *nn.ParameterList*. Hence, I reworked the code to make parallel training possible and also developed my own FourierFlow library that includes both FNO and FFNO architectures. If you're interested in trying them out, feel free to check:

https://github.com/VikVador/FourierFlow

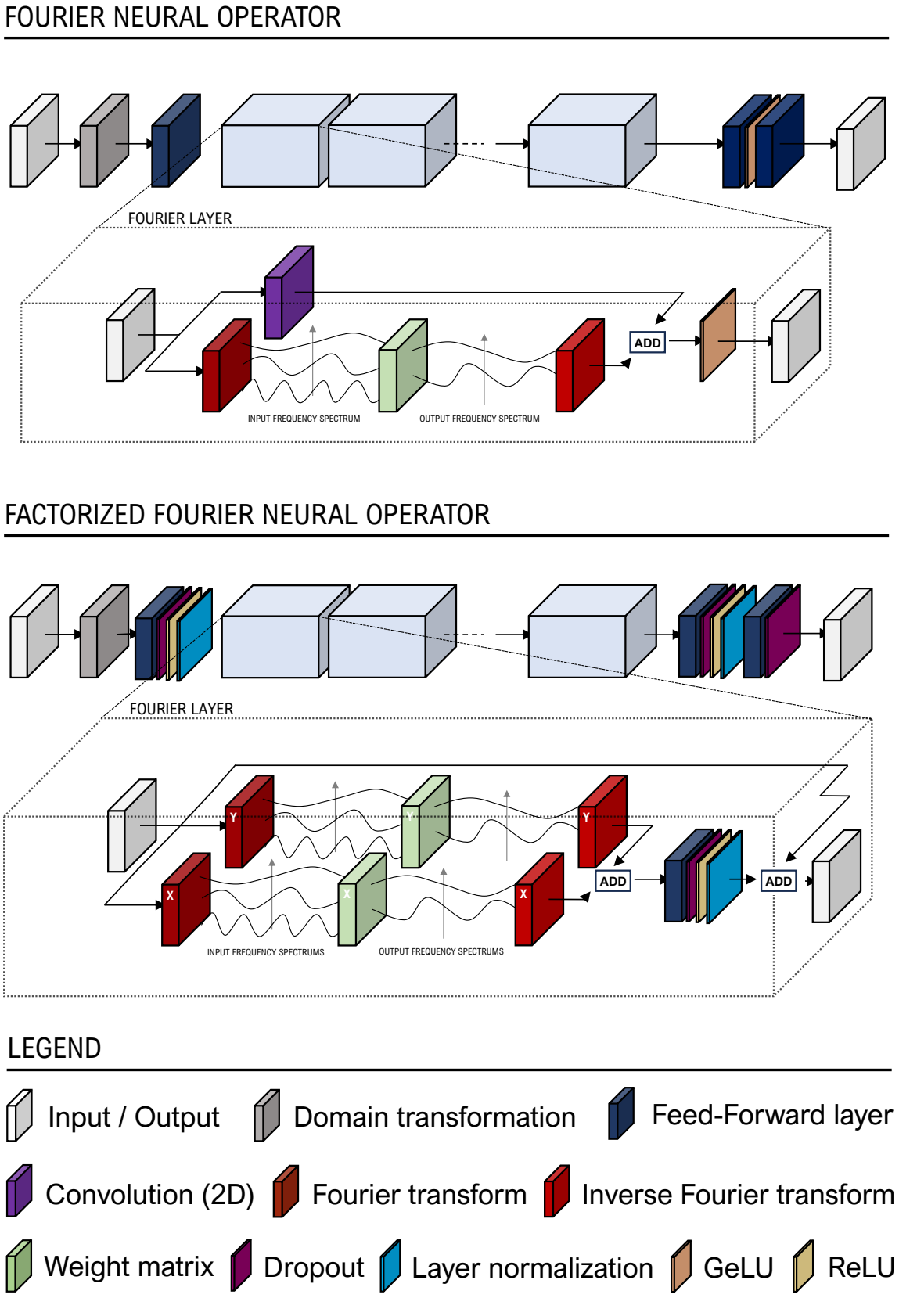


Figure 17: Vizualisation of the Fourier Neural Operator (FNO) and Factorized Fourier Neural Operator (FFNO) architectures. In the figure, **ADD** means the addition of the inputs along the batch dimension.

Chapter Four

Datasets

The aim of this chapter is to clarify the type of flow under investigation, outline the general dataset generation procedure, describe the processes of coarsening and filtering, define the simulation parameters and their values, and detail the datasets generated for this study.

4.1 INTRODUCTION

In the world of machine learning, as for any scientific fields involving numerical simulations, the adage "garbage in, garbage out" stands strong. Indeed, a model, even if its the latest state of the art available right now, can only be as proficient as the quality of data it learns from. Therefore, the importance of data quality cannot be overstated, as it creates the fundation upon which models are built.

This work uses the quasigeostrophic flow Python numerical solver, known as *PyQg*, to generate datasets. Especially, this approach becomes imperative due to the limitations encountered when attempting to use real data, such as satellite imagery of the oceans. As a matter of fact, the complexities of space and time sampling introduce formidable challenges that makes the use of such data impractical and insufficient for effective model training. Indeed, space and time sampling challenges are the result of the nature of the ocean which can be simply described as huge, dynamic and full of physical phenomenons occuring at different space and time scales. Satellite imagery, while a powerful tool, introduces biases arising from limited spatial coverage and temporal resolution. Therefore, the PyQG numerical solver seems to be the most simple and efficient solution.

In the mid-latitude region, the primary focus area, two distinct flow types govern the range of possible flows found there. The first one is a jet-driven flow, where geostrophic equilibrium predominates, resulting in structured, linear flow. Conversely, the second scenario involves eddy-driven flows, where the strength of inertial forces disrupts the geostrophic equilibrium, emphasizing flow speed. In this context, the previously straightforward flow path transforms into a more turbulent dynamic, giving rise to the generation and dispersion of eddies across the region.

In their studies (Bolton and Zanna, 2019; Ross et al., 2023; Zanna and Bolton, 2020), the adopted approach centers on concentrating on a single flow type, deriving a parameterization, and then assessing its applicability to other flows. They are several reasons that could justify this choice. Indeed, the decision to focus training on a single flow type is grounded in practical considerations driven by real-world limitations and the complexity of data acquisition.

In practical situations, the acquisition of comprehensive real data remains challenging. Focusing on a specific flow type aligns with this scarcity of data, adding practicality to the study. Successfully deriving a parameterization for this selected flow type holds substantial potential, showcasing the ability to create effective representations using a constrained dataset. Additionally, the generation of data involves inherent costs. Within our framework, the use of an idealized model simplifies Navier-Stokes equations and considers solely two fluid layers. This simplification enables more manageable solutions and supports academic exploration and concept development. Therefore, transferring this approach to a more realistic context introduces the challenge of limited data due to the complexities of data collection. As a result, the need to train with a modest dataset arises from these practical constraints.

Nevertheless, owing to the simplicity of our model and their work leaving the exploration of the impact of more complex datasets for future research, our attention will be aimed at investigating the effects of dataset size and complexity while also conducting tests involving the new Fourier Neural Operator architectures for the parameterization of subgrid scale processes. Our study will contain datasets that range from single-simulation datasets to those that consist of multiple simulations of a single flow type, and will extend even further to datasets incorporating samples from both types.

4.2 GENERATING A DATASET

The dataset generation, which includes flow field quantities like velocity fields and potential vorticity, along with the corresponding missing subgrid process contributions as defined in Eq.21, 22 and 23, is achieved as follows:

1. **High-resolution simulation**: A simulation is run for a duration of 10 years with a one-hour time step, using a specified set of initial and boundary conditions denoted as B. There are 2 main reasons explaining the need of doing such a long-time simulation:

    - From a physics point of view, the solution to the quasigeostrophic problem eventually settles into a quasi-steady state. This means that once this equilibrium state is reached, the physical variables that describe the flow will fluctuate around a certain value. To tackle such challenges, like many computational fluid dynamics solvers, *PyQG* uses an iterative approach. It begins with initial conditions, then refines the solution at each step until the system reaches a quasi-steady state.

      Between the early stages of the simulation and the moment where the quasi-steady state is achieved, the solution is referred to as the "transient solution". This phase captures the transformation of the solution from its initial state to the final one that accurately depicts the intended flow. This transient solution can persist for a considerable number of time steps, highlighting the need to allow the simulation to run for an adequate duration to reach the quasi-steady state. Otherwise, if the simulation is sampled prematurely, the obtained results might lack genuine physical meaning.

- In atmospheric flows, even the faster scales like those of small eddies evolve over relatively long time spans, typically a few days. Consequently, to efficiently sample the simulation, a significant amount of time needs to pass between two consecutive samplings. To gather just a few thousand samples, it is necessary to run the simulation for around 10 years. This extended runtime ensures the generation of a diverse range of samples that accurately describe the flow dynamics.

2. **Low resolution simulation**: Under the same initial and boundary conditions B, a simulation is conducted for 10 years, using a one-hour time step.

3. **Sampling**: Starting after 4 years (for an eddy-driven flow) and 6 years (for a jet-driven flow) of simulation, potential vorticity is sampled every 1000 hours until the simulation concludes. Indeed, as explained in chapter two, this variable serves as the prognostic factor from which all other physical flow quantities can be derived.

4. **Extracting subgrid processes contributions**: First of all, one needs to assume that the period of time chosen for simulation is long enough to sample efficiently but short enough for both high- and low-resolution simulations to remain correlated. As a reminder, the main issue that one wants to correct is the energy defficiency of the low-resolution simulation. It is the energy that defines the dynamic of a simulation, therefore if the simulation is run for too long, the energy defficiency will at some point affect the dynamic of the simulation and make it diverge from one another regarding the physics of the flow they aim to describe.

   Assuming this assumption holds, the next steps involve coarsening and filtering the high-resolution simulation. For each sample, it is necessary to reduce the resolution and smooth out the solution obtained from PyQG for the potential vorticity. The resulting solution serves as a condensed representation of the high-resolution simulation. While not as visually precise, it accurately maintains the energy budget—a contrast to the deficient energy dissipation in the low-resolution counterpart due to neglected small-scale processes.

   Finally, the difference, for a given flow quantity, between the coarsened and filtered low-resolution simulation and the original low-resolution simulation reveals the neglected contribution of to subgrid-scale processes.

4.2.1 COARSENING AND FILTERING

Coarse-graining involves reducing the resolution of a simulation or dataset while retaining essential details. This technique includes averaging values over larger regions to effectively downscale the resolution and capture the behavior of larger-scale processes. The procedure is straightforward: begin with a high-resolution simulation and divide the grid into larger blocks matching the lower resolution grid size. Fininally, within each block, average (or aggregate) the values.

Filtering modifies the frequency content of a signal or dataset by eliminating unwanted noise, high-frequency variations, or fine-scale details while maintaining significant features.

Though coarsening alone offers a way to achieve lower resolution data, it might not tackle noise or unwanted high-frequency variations effectively. On the contrary, filtering could result in a smoother data representation but might miss out on large-scale features captured by coarser resolution. Combining both filtering and coarsening in certain cases can yield more informative outcomes, producing a cleaner, smoother depiction of large-scale behavior while retaining essential features.

Various methods can be used to perform coarsening and filtering operations on datasets. However, it is crucial to be cautious, as the approach to extracting subgrid processes contributions can impact both the training quality and predictive ability of the neural network. This aspect, namely evaluating parameterization quality based solely on coarsening and filtering choices, has already been explored by (Ross et al., 2023), and we will adopt their recommended approach for our study. To achieve coarsening, we will use a method called **spectral truncation**. This involves removing modes from the high-resolution simulation to match the available modes in the low-resolution simulation. Once truncated, the grid values are then averaged over larger regions corresponding to the low-resolution grid. In the case of filtering, a sharp filter is applied afterwards. This filter preserves low frequencies while reducing higher frequencies above a specified threshold. Both of these operations are already integrated into PyQG, and further details on their functioning can be found in the PyQG documentation on its github.io page.

4.3 SIMULATIONS

In order to run a simulation, several parameters must be chosen:

4.3.1 SOLVER

The high-resolution simulation employs a grid size of 256×256 pixels, while the low-resolution simulation uses a grid of 64×64 pixels. Both simulations run for a duration of $T = 10$ [years], with a time step of one hour. Sampling begins after the system has achieved a quasi-steady state solution, which takes 4 years for eddy-driven flows and 6 years for jet-driven flows.

4.3.2 MODEL

Each model consists of a doubly-periodic square domain with dimensions $L = 10^6$ [m], flat topography, and a combined depth of $H = H_1 + H_2 = 2500$ [m]. A constant zonal velocity shear, $\Delta U = U_1 - U_2 = 0.025$ [m/s] is imposed between the upper and lower layers ($U_1 = 0.025$ [m/s], $U_2 = 0$). The deformation radius r_d, a key measure for baroclinic instability and mesoscale turbulence, is set at 15000 [m], with the requirement that $r_d/\Delta x > 2$ for effective mesoscale eddy resolution. For instance, a 256×256 grid with $\Delta x_{\text{hires}} = L/256 = 3906.25$ [m] leads to $r_d/\Delta x_{\text{hires}} = 3.84$, effectively resolving mesoscale turbulence. Conversely, a $\Delta x_{\text{lores}} = L/64 = 15625$ [m] grid results in $r_d/\Delta x_{\text{lores}} = 0.96$, rendering the simulation unrealistic. Lastly, the heights of the fluid layers are determined, with $H_1/H_2 = 0.25$ for an eddy-driven flow and $H_1/H_2 = 0.1$ for a jet-driven flow.

All the previously mentioned parameters will remain constant throughout this study. Consequently, only two parameters can be adjusted to tweak the flow dynamics. The first parameter is the bottom drag coefficient r_{ek}, which determines the influence of frictional forces in the bottom layer. Additionally, there is the slope of the Coriolis parameter β, altering the intensity of the Coriolis force with altitude. Increasing its value amplifies inertial forces, leading to an eddy-driven flow. In the work of Ross et al., 2023, they chose to use $r_{ek} = 5.789 \times 10^{-7}$ $[s^{-1}]$ and $\beta = 1.5 \times 10^{-11}$ for the eddy-driven flow, while for the jet-driven flow, $r_{ek} = 7 \times 10^{-8}$ $[s^{-1}]$ and $\beta = 1 \times 10^{-11}$ were used. The resulting flows are depicted in the Fig.18 and correspond to the expected behavior of eddy-driven and jet-driven flow types.

4.4 GENERATED DATASETS

Throughout this study, various dataset configurations were explored to enhance the performance of the well-established FCNN model and to assess, for the first time, the capabilities of the Fourier Neural Operators in the context of subgrid processes parameterization. Just like the reference papers we built upon, we generated datasets comprising samples from a single flow type. Additionally, we explored the approach of using a dataset that contains samples from multiple simulations of the same flow type. The idea is that both flow types exhibit distinct dynamics, leading to orders of magnitude differences in characteristic flow variables such as potential vorticity. Thus, by training the neural network on samples that maintain the same flow dynamic while varying its intensity, like changing flow speed, makes it learn the dynamics of the flow while experiencing a broader input value spectrum that shares some values closer to the one of the other flow.

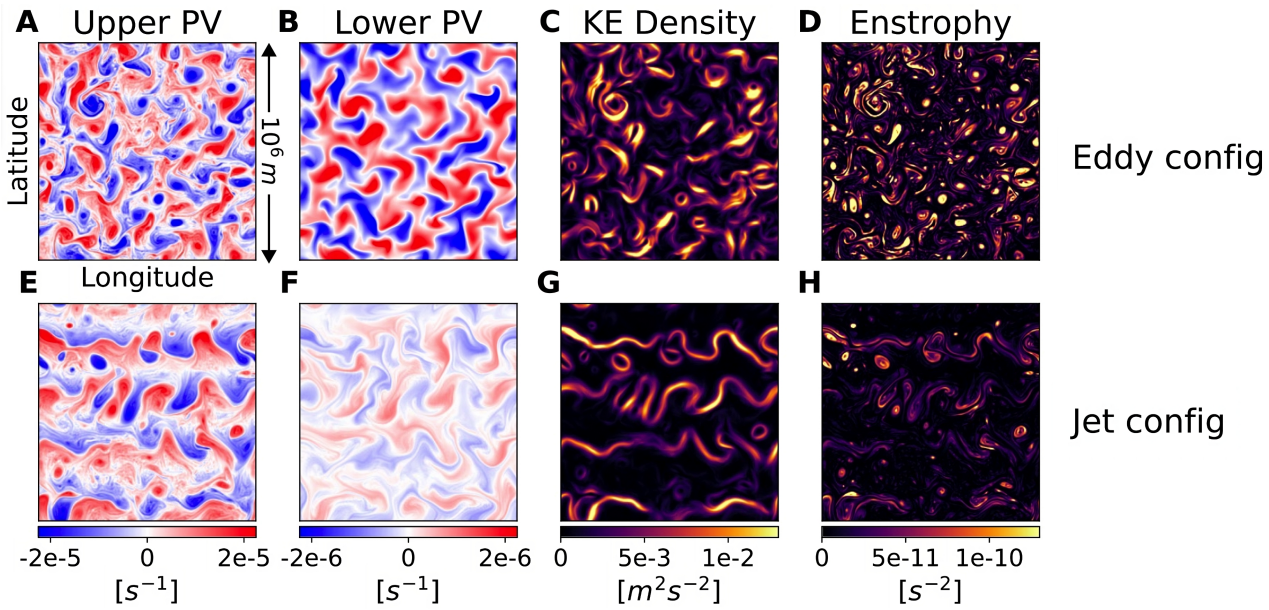


Figure 18: Snapshots of upper (A,E) and lower (B,F) potential vorticity (PV), barotropic kinetic energy (C,G), and barotropic enstrophy (D,H) for simulations run in eddy (A-D) and jet (E-H) configurations over a square, doubly-periodic domain of length 10^6 m. Eddy configuration results in an approximately isotropic distribution of vortices, while jet configuration results in the formation of stable, long-lived jets with more coherent latitudinal structure. This figure and caption comes from Ross et al., 2023.

To achieve this, simulations were conducted using bottom drag coefficient and coriolis parameter slope values drawn from uniformly distributed distributions within +-5% of the values used by Ross et al., 2023. This value was found empirically and could potentially be increased, but around 20% (- for eddies and + for jets), the dynamic differences of both flow began to diminish. Lastly, the ultimate datasets consist of multiple simulations of both flow types. Here, our interest lies in determining whether the neural network can effectively derive a parameterization that applies to both types of flows using the combined data. We want to explore whether the network, even without full training on one flow type, can still enhance its performance for that type by incorporating features learned from the other.

Our study will be organized into 6 phases. At times, the focus will be on observing the impact of changing the dataset type on the results. In other instances, the emphasis will be on maintaining the dataset type while exploring the effects of increasing the number of samples. The datasets used in each phase are summarized in Tab.1.

Phase	**Name**	**Type**	**Flow(s)**	**Nb. Samp.**	**Nb. Sim.**	**Samp. per Sim.**
1	UE5000	UNIQUE	E	5000	1	5000
-	UJ5000	UNIQUE	J	5000	1	5000
2	ME5000	MIXED	E	5000	10	500
-	MJ5000	MIXED	J	5000	10	500
3	ME10000	MIXED	E	10000	20	500
-	ME20000	MIXED	E	20000	40	500
-	MJ10000	MIXED	J	10000	20	500
-	MJ20000	MIXED	J	20000	40	500
4	F5000	FULL	E & J	5000	10	500
-	F10000	FULL	E & J	10000	20	500
-	F20000	FULL	E & J	20000	40	500
-	F40000	FULL	E & J	40000	80	500
5	F5000	FULL	E & J	5000	10	500
6	F5000	FULL	E & J	5000	10	500

Table 1: Table summarizing the names and characteristics of all the datasets. In the case of the **full** datasets, the proportions are 50% eddies (E) and 50% jets (J). For the unique datasets, the r_{ek} and β values chosen are the one of Ross et al., 2023. In the case of the *UE5000* and *UJ5000* datasets, r_{ek} and β are the one used in Ross et al., 2023

Chapter Five

The story of the results

This chapter provides a comprehensive overview of the research methodology, covering the setup details and metrics used for model evaluation. The results will be presented in six distinct phases to enhance clarity and emphasize key findings.

5.1 METHODOLOGY AND SETUP

5.1.1 BASELINES

To evaluate the performance of a newly learned parameterization, a comparison against baseline models is essential. Thus, the three different configurations of the fully convolutional neural network introduced in the work of Ross et al., 2023 will serve as baselines, as they represent their best results achieved. Each of these models was trained on a dataset containing 5000 samples from a single eddy-driven simulation (*UE5000*, as shown in Tab.18), during 50 epochs. The input for all networks is the potential vorticity q, and each network outputs one of the formulations of the subgrid term (as described by Eq.21, 22, 23). These neural networks will be used for both offline and online testing. Additionally, beyond aiming to improve upon their results, it is important to verify whether using their architecture and the same dataset while sampling differently produces consistent results, providing reassurance regarding our setup.

In addition to these neural network models, several well-known analytical parameterizations from the world of turbulence closure modeling, including the one proposed by Zanna and Bolton, 2020, have also been included in the comparison. However, they will solely serve as baselines for the online testing phase. This decision is based on practical reasons: introducing these parameterizations would significantly increase the number of experiments and extend the already substantial computation time required for the extensive set of experiments already prepared. Furthermore, while integrating these parameterizations into their benchmark framework (Ross et al., 2023) was relatively straightforward for online testing, the same ease did not apply to the offline testing phase.

The first parameterization under consideration is the one of Smagorinsky (Smagorinsky, 1963), a widely used turbulence closure model in numerical weather modeling. This approach estimates unresolved turbulent stresses by introducing an eddy viscosity term determined by the resolved flow variables, specifically the derivatives of the velocity fields. The model incorporates a constant parameter known as the Smagorinsky coefficient C_S, which plays a crucial role in establishing the eddy viscosity.

Although computationally efficient, the model accuracy can be influenced by the chosen Smagorinsky coefficient and might not comprehensively capture intricate turbulent flows. By taking into account the approach used by Ross et al., 2023, the C_S value is set at 0.15 for this study.

Before introducing the second parameterization, it is crucial to understand an important concept related to geostrophic turbulence. Indeed, this turbulence drives the transfer of enstrophy (measure of the magnitude of vorticity squared) to smaller scales, as outlined by Charney, 1971.

As an example, when one talks about geostrophic turbulence transferring enstrophy to smaller scales, it means that as the turbulence occurs, this swirling motion gets distributed to smaller and smaller regions within the fluid. Another way to think about this, one can imagine stirring a cup of coffee with a spoon. The swirling motion created initially is like the enstrophy, and if one keeps stirring, that swirling motion gets spread out throughout the coffee, especially in smaller whirlpools. Similarly, in geostrophic turbulence, the spinning or swirling energy gets spread to smaller areas within the fluid.

Therefore, in a numerical simulation, managing enstrophy dissipation near the grid-scale is essential to prevent its accumulation. To address this, a common strategy involves using a horizontal hyper-viscosity, frequently biharmonic, i.e. an additional term of fourth order is added to the partial differential equations describing the flow. This viscosity selectively dissipates enstrophy near the grid-scale, much like the Smagorinsky approach. However, a complication arises at resolutions close to the scale where smaller eddies emerge. In such cases, these closures not only dissipate enstrophy but also a notable portion of the total energy. This is problematic since geostrophic turbulence channels energy to larger scales while transferring enstrophy to smaller scales. Hence, an ideal parameterization should prioritize enstrophy dissipation without dissipating energy at small scales.

To tackle this issue, Jansen and Held, 2014; Jansen, Held, et al., 2015 proposed a novel approach introducing a class of subgrid parameterizations that dissipate enstrophy while preserving most of the energy. The method involves combining a standard hyperviscous closure (ensuring enstrophy dissipation) with a mechanism to return the dissipated energy back to the resolved flow at larger scales which they called backscaterring. By redirecting the energy to larger scales, this parameterization maintains a more realistic energy cascade, resulting in more energetic eddy fields. Their parameterization, that will be named Backscatter and Biharmonic Dissipation, has two hyper-parameters, C_B (fraction of Smagorinsky-dissipated energy scattered back to larger scales) and C_S^2 (dissipation constant), which are respectively set to 1.2 and 0.007. Once again, this choice is based on the results obtained in Ross et al., 2023.

Lastly, the final parameterization used as a baseline is the one obtained from data using relevance vector machine with an idealized primitive equation model presented in Zanna and Bolton, 2020. The set of weights used is the one derived from their human-in-the-loop technique during the regression process. If the reader is seeking more detailed explanations about these three analytical expressions, they can be found in the appendix *a*, *b* and *c* of the work by Ross et al., 2023.

5.1.2 INVESTIGATED ARCHITECTURES

In total, four neural networks were tested extensively throughout this work. Indeed, the first one is the **fully convoslutional neural network**, the purpose is to improve upon the best results obtained in Ross et al., 2023. Additionally, a **U-NET**, a state-of-the-art model for convolution operations on spatiotemporal data, is included. Furthermore, the first **Fourier Neural Operator** (FNO) and its improved architecture called the **Factorized Fourier Neural Operator** (FFNO) are also tested. The latter two networks operate primarily in the spectral domain, making it interesting to compare their results with the first two networks operating in the original space-time data representation.

A detailed visualization of all these networks is provided in Fig. **??** and **??**. Additionally, comprehensive documentation detailing the network architecture parameters settings is available in the code library associated with this thesis, which will be accessible at the end page of this work. Specifically, within the file *neural_networks.py*, one can find the configurations for all these networks along with the corresponding values used.

It is important to note that, to ensure a fair comparison, each of these neural networks has a similar number of trainable parameters, roughly around 300,000. These parameter configurations will remain consistent from phase one to five of this study. However, a thorough analysis of the FFNO architecture will be conducted in phase six, with each tested combination documented comprehensively to provide clarity on the choices made.

5.1.3 INPUTS

The input, in the case where only one flow field variable is used, is represented by a 2-dimensional matrix with dimensions $N \times N$, where $N = L/\Delta x$, and L denotes the domain size while Δx refers to the spatial resolution. The possible flow field variables that can serve as inputs include the horizontal velocity u, vertical velocity v, and potential vorticity q. Additionally, any combination of these variables can be provided as input. However, for the sake of consistency and to manage the number of configurations efficiently, the possible test cases were limited to the following combinations:

$$q,\ (q,\ u),\ (q,\ v),\ (q,\ u,\ v) \tag{24}$$

5.1.4 OUTPUTS

The output of the parameterization can be one of three possible formulations: the **total subgrid forcing** $S_{q_{tot}}$ (see Eq. 21), the **subgrid forcing of potential vorticity** S_q (see Eq. 22), and the **subgrid flux** $\mathbf{\Phi}_q$ (see Eq. 23). Although these formulations are highly correlated, only the last one of them ensures that the conservation law is respected. Indeed, when the neural network is trained to predict the subgrid flux, the divergence operation is numerically and internally computed by *PyQG* afterwards.

Ensuring the conservation law is respected is of great importance. In the first two configurations, there is no guarantee that the neural network will predict a quantity resulting from the divergence of another.

Nevertheless, the divergence operator serves a crucial role in diffusing physical quantities, smoothing out information and preventing localized accumulation. Simulations that do not adhere to conservation laws can exhibit unpredictable behavior, potentially leading to energy peaks, simulation instability, and divergence due to energy explosions.

For the first two subgrid term formulations, Ross et al., 2023 discovered that ensuring the output of the last layer has a zero mean significantly improved offline results and stabilized online simulations. Consequently, this operation will be applied to the output when the neural network uses either of the first two formulations. However, this feature will be turned off for the conservative formulation.

Lastly, it is important to note that when the chosen formulation is the subgrid flux $\mathbf{\Phi}_q$, the neural network outputs not one but two values. Specifically, it calculates the subgrid flux values in both the x- and y-directions. Therefore, during offline tests, the mean squared error (MSE) is calculated individually for both of these quantities. However, for simplicity, as in Ross et al., 2023, the resulting MSE error discussed throughout this work concerning the subgrid flux is the average of the MSE errors computed for both quantities separately.

5.1.5 DATASETS

The datasets used for training the neural networks in each phase, along with their respective short names, are summarized in Tab.1. The summarizing table of offline results for each phase also includes the short name of the corresponding dataset used. In addition to the training datasets, six additional datasets to evaluate the quality of our parameterizations were created.

For offline testing, three datasets are used each composed of samples of: eddy-driven flows (**eddies offline**), jet-driven flows (**jets offline**), and another dataset containing samples from both flow types (**full offline**). Each of these datasets consists of 5000 samples collected from 10 different simulations (with 5 from each type in the full offline dataset). Additionally, three datasets for online testing were also created: eddy-driven (**eddies online**), jet-driven (**jets online**), and samples from both flow types (**full online**). However, by contrast to the offline datasets, they not only include subgrid scale processes values but also contains the energy spectrum of the flow, values at each timestep of all flow quantities, and more.

5.1.6 TRAINING

The training conditions used remain consistent with those described in Ross et al., 2023. Specifically, the batch size is set at 64 samples, the learning rate (γ) is equal to 0.001, the optimizer is ADAM, and the scheduler is multi-step with milestones set at (4/8), (6/8), and (7/8) of the training. The number of epochs is 50, and the loss function used is the mean squared error between the predicted subgrid scale process contribution and its true value. In phase 5, we will dive deeper into exploring the FFNO prediction capabilities by testing various combinations of training setups. These combinations and their descriptions will be presented accordingly in phase 5.

5.2 METRICS

To assess the quality of the newly learned parameterization, a benchmarking framework was developed by Ross et al., 2023. This framework enables us to evaluate the parameterization performance through various testing methods.

5.2.1 OFFLINE

Offline metrics serve the purpose of assessing how well the parameterization predicts its target. For each fluid layer, like the upper and lower layers shown in Fig.12, two metrics are used. The first metric is the **coefficient of determination** R^2, expressed as:

$$R^2 = 1 - \frac{\mathbb{E}\left[(S - \hat{S})^2\right]}{\mathbb{E}\left[(S - \mathbb{E}[S])^2\right]} \tag{25}$$

Here, S represents the exact subgrid scale contribution, $\hat{S}$ is the prediction made by the parameterization, and $\mathbb{E}$ stands for statistical expectation. This metric interpretation is simple: it yields a value of 1 for perfect predictions, 0 when predictions are no better than always guessing the mean, and negative values when predictions are worse than guessing the mean. The second metric is the **Pearson correlation** (ρ), given by:

$$\rho = \frac{\mathrm{Cov}(S, \hat{S})}{\sigma_S \sigma_{\hat{S}}} \tag{26}$$

Here, σ represents the empirical standard deviation of a quantity across the dataset. Pearson correlation varies between -1 and 1 and can remain high even if R^2 is negative. For example, if predictions consistently have a wrong but proportional scaling factor. To compute these metrics, the procedure is straightforward: at each time step, the metric value is calculated for each pixel and then averaged. The final results are obtained by averaging these values across all time steps.

5.2.2 ONLINE

The online metric serves the purpose of evaluating the parameterization performance in real-time simulations. The approach involves conducting low-resolution simulations and applying the parameterization at each time step to correct the simulation. Then, various metrics can be calculated on these simulations to determine if they exhibit meaningful physical behavior and whether they align with the results of high-resolution simulations that share the same initial and boundary conditions.

It is crucial to note that achieving good results in the offline phase does not necessarily guarantee success in practical scenarios. The benchmarking framework possess various metrics to assess the quality of the newly developed parameterization. However, using all of these metrics would generate a vast amount of data that could be too time consuming (alone) to analyze.

Therefore, the selected metrics for evaluation are:

- **Spectrum analysis**: For energy-related flow quantities, it is possible to extract and examine their power spectrum in spectral space. This allows us to see the contributions from different scales in the spectrum. The goal is to determine whether the spectrum of a low-resolution simulation, corrected with a parameterization, matches that of the high-resolution simulation. The specific quantities of interest are:

  1. **KEflux** (Kinetic Energy Flux): This reveals how kinetic energy is being transferred across different lengthscales.
  2. **KEfrictionspec** (Friction Energy Spectrum): Indicates the amount of energy lost due to bottom drag, occurring between fluid layers or at the ocean floor, for each lengthscale.
  3. **APEflux** (Available Potential Energy Flux): Measures the potential energy available for transfer between different lengthscales.
  4. **APEgenspec** (New Available Potential Energy Spectrum): Represents the newly generated potential energy at each lengthscale.

  While these metrics might seem complex, the main focus should be on one key idea: these metrics capture different aspects of how energy moves within a fluid, which is essential for defining flow dynamics. If one observes matching spectra between simulations, it confirms that the parameterization is effectively redistributing and addressing energy deficiencies in the simulation. These four metrics can be computed for both layers of the simulation, but for clarity, only the spectrum for the first layer will be presented (this is an arbitrary choice for the sake of conciseness).

- **Differences between time-averaged power spectra and fluxes**: While spectrum analysis provides a useful visual tool, it is also valuable to quantitatively measure the differences between power spectra. This can be accomplished by calculating the differences between time-averaged power spectra and fluxes. To compute this difference, let f represent the power spectrum curve of a specific quantity. The spectral difference can be calculated as follows:

$$\text{spectral\_diff}(\text{sim}\,1, \text{sim}\,2; f) \equiv \sqrt{\frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \left(f_{\text{sim}\,1}(k) - f_{\text{sim}\,2}(k)\right)^2} \tag{27}$$

  Here, $\mathcal{K}$ represents a set of isotropic wavenumbers common to both simulations. In our case, $\mathcal{K}$ is evenly distributed in logarithmic space and covers up to $2/3$ of the Nyquist frequency of the low-resolution simulation, as described in Ross et al., 2023.

- **Differences between spatially flattened probability distributions**: Even if the energy spectrum appears satisfactory, it does not necessarily mean that the dynamics represented by the corrected low-resolution simulation will match those of the high resolution. Hence, it is valuable to compute the empirical distributions of various flow quantities at the end of the simulation.

This can be done using the earth mover's distance (or, in other words, the Wasserstein distance with a p value set to 1):

$$\text{distrib\_diff}(\text{sim}\,1, \text{sim}\,2; f) \equiv \int_{-\infty}^{\infty} |P_{\text{sim}\,1}(f \leq x) - P_{\text{sim}\,2}(f \leq x)| \, dx \tag{28}$$

Here, $P_{\text{sim}}(f \leq x)$ represents the cumulative distribution function of quantity f in a given simulation. Imagining the two probability density functions as mounds of earth, this metric corresponds to the minimum amount of work required to move all the mass from one mound to the other. For 1-dimensional distributions, it reduces to the integral of the difference in each cumulative distribution function (which we empirically approximate). These differences are calculated for the quasi-steady-state distributions (marginalized over space and at the final timestep) of u, v, q, kinetic energy density, and enstrophy int the first layer (once again, this a choice for conciseness).

It is important to notice that when comparing low-resolution to high-resolution metrics, we are comparing the distributions of quantities like u (solution of the high-resolution) and $\bar{u}$ (solution of the low-resolution), so histograms are properly normalized, as outlined in Ross et al., 2023.

- **From difference to similarity**: Defining various distance metrics can lead to challenges when comparing them, especially due to differences in units. However, the actual value of a metric is not the main concern. What matters is whether the result for parameterized simulations line up to the one of high-resolution simulations more than a simple low-resolution simulation. To address this, the distance metrics are transformed into similarity scores, indicating how close parameterized models are to both type of simulations:

$$\text{Similarity (param, high-res; diff)} \equiv 1 - \frac{\text{diff(param, high-res)}}{\text{diff(low-res, high-res)}} \tag{29}$$

This similarity score is approximately 1 if the parameterized model distance to a high-resolution is much smaller than that of the low-resolution model (and exactly 1 for the high-resolution model), approximately 0 if this distance is about the same as the low-resolution model (and exactly 0 for the low-resolution model), and less than 0 if the distance is larger.

- **A simple but meaningful visualization**: While numerical metrics provide insight, observing potential vorticity distribution offers another way to assess the parameterization quality. For the final time step of the simulation, we will present potential vorticity plots for high-resolution, low-resolution, neural networks, analytical baselines, and our parameterizations. This practical assessment complements the other rigorous metrics. After all, a good graphic can often speak volumes, don't you think?

PHASE 1

In this first phase, the neural networks are trained using a single flow simulation consisting of 5000 samples (*UE5000* and *UJ5000*). The idea is to train them under the same conditions as those in Ross et al., 2023 to detect any potential differences in results.

Offline

First and foremost, it is clear that our baseline neural networks, using identical architectures, training conditions, and simulations but with varied sampling, already show a difference. Specifically, the offline results of the FCNN trained on *UE5000* exhibit a reduction of 3% to 6% in ρ and R^2 measurements for both layers.

A recurring observation is that for all architectures trained on *UE5000*, the offline results excel on the eddies offline dataset (not shown) but perform poorly on the jets online datasets, leading to negative R^2 scores. However, when trained on *UJ500*, neural networks tend to perform slightly worse on the eddies offline dataset but slightly better on the jets offline. In other words, the results are polar opposites in the first scenario, while in the second scenario, both results are unsatisfactory.

Whatever the situation, it is evident that predicting the subgrid flux $\mathbf{\Phi}_q$ consistently yields the best results, and this is also true when neural network inputs combine potential vorticity and velocity fields. The Tab.2 provides the best results for each architecture type.

The most intriguing observation is that the FFNO, whether trained on eddies or jets, achieves a positive R^2 in the lower layer when tested on jet-type flows. This contrasts with the baseline parameterizations, in their study (Ross et al., 2023), they struggled to find a neural network parameterization that achieved positive results in this specific layer.

Online

The energy spectrum, visible in Fig.19 and 20, shows differences between FCNN and UNET. Indeed, the FCNN fails to capture the kinetic energy flux spectrum, while the U-NET aligns better with the high-resolution simulation spectrum shape.

However, this result pales in comparison to the FFNO, which already demonstrates strong performance in predicting both spectra for both eddy and jet-driven online testing datasets. Nevertheless, FNO faces challenges, with only the one trained on **UE5000** managing to produce a stable eddy-driven online simulation; in the other three cases, simulation stability criterion (CFL number) was violated.

Moving to the distribution similarities shown in Fig.21 and 22, one can observe that in the space-time domain, the baseline neural networks and U-NET are more capable of maintaining simulations with dynamics less divergent from the original. However, neither FNO nor FFNO achieves a correct spatial distribution of flow quantities. Lastly, even though FFNO demonstrates strong offline performance, a quick look at Fig.24 reveals that its simulation flow dynamics significantly deviate from the high-resolution.

Conclusion

In summary, varying the sample approach while maintaining consistent neural network architectures and training conditions has a noticeable impact. The baseline FCNN trained on *UE5000* displays a reduction of 3-6% in ρ and R^2 for both layers. Furthermore, architectures trained on *UE5000* perform well in eddies offline but poorly in jets online, resulting in negative R^2 scores.

However, when trained on *UJ500*, neural networks produce slightly worse in eddies and slightly better in jets offline testing. Predicting subgrid flux $\mathbf{\Phi}_q$ consistently yields optimal results. Particularly, FFNO stands out by achieving positive R^2 for jet-type flows in the lower layer, unlike the baseline parameterizations.

Transitioning to the online phase, FFNO excels in predicting energy spectra for both eddy and jet-driven testing datasets, while FCNN faces challenges. Distribution similarities uncover that baseline neural networks and U-NET better preserve a bit more the original dynamic, while FNO and FFNO struggle with spatial flow quantity distributions. Indeed, despite robust offline performance, FFNO simulation dynamics substantially differ from high-resolution counterparts.

	Dataset	Input	Output	Loss (Val.)	ρ_1	ρ_2	R_1^2	R_2^2
FCNN	UE5000	(q, u, v)	ϕ_q	0.05	0.99	0.97	0.97	0.95
-	-	-	-	-	0.94	0.76	-0.21	-10.63
-	UJ5000	(q, u, v)	ϕ_q	0.22	0.97	0.83	0.72	0.29
-	-	-	-	-	0.95	0.85	0.88	0.72
UNET	UE5000	(q, u, v)	ϕ_q	0.35	0.91	0.81	0.82	0.64
-	-	-	-	-	0.83	0.52	-0.69	-16.34
-	UJ5000	(q, u, v)	ϕ_q	0.77	0.80	0.61	0.54	0.19
-	-	-	-	-	0.83	0.56	0.65	0.19
FNO	UE5000	(q, u, v)	ϕ_q	0.80	0.64	0.55	0.36	0.19
-	-	-	-	-	0.58	0.18	0.18	-0.92
-	UJ5000	(q, u, v)	ϕ_q	2.84	0.19	0.14	-1.33	-0.93
-	-	-	-	-	0.45	0.15	-0.28	-1.80
FFNO	UE5000	(q, u, v)	ϕ_q	0.22	0.91	0.88	0.83	0.78
-	-	-	-	-	0.88	0.55	0.76	0.12
-	UJ5000	(q, u, v)	ϕ_q	0.65	0.84	0.47	0.70	0.21
-	-	-	-	-	0.89	0.57	0.79	0.30
FCNN	UE5000	q	$S_{q_{\text{total}}}$	0.09	0.91	0.95	0.82	0.90
-	-	-	-	-	0.86	0.82	-0.52	-4.63
FCNN	UE5000	q	S_q	0.09	0.91	0.95	0.82	0.90
-	-	-	-	-	0.86	0.82	-0.52	-4.63
FCNN	UE5000	q	ϕ_q	0.05	0.98	0.97	0.97	0.94
-	-	-	-	-	0.95	0.76	-0.22	-8.07

Table 2: | **Offline - Phase 1** | Summarizing table of offline results. The white lines represent results for the **eddies offline** dataset, while the colored lines correspond to the **jets offline** dataset. Along with the training conditions, the validation loss (calculated on an unseen dataset of 5000 samples; for eddy training, the corresponding dataset is named **eddies validation**, and likewise for jets and the full dataset) is shown. The metric values for R^2 (see Eq.25) and ρ (see Eq.26) are given for both the upper layer (i = 1) and lower layer (i = 2).

PHASE 2

In the second phase, neural networks are trained on a dataset containing samples from 10 distinct simulations of a single flow type *ME5000* and *MJ5000*). The idea is to maintain the characteristic flow dynamics while expanding the input value range, which could enhance the parameterization ability to generalize to the other type of flow.

Offline

As shown in Tab.3, there is improvement in every metric score. Notably, the FFNO performance stands out: when trained on the mixed eddie-driven flow dataset (*ME5000*), its overall results on the eddies offline dataset are even worse than the baselines. However, although the results in the upper layer are still lower than the baselines on the jet offline dataset, the results of the lower layer clearly surpass the baselines. In addition to being positive, it reaches the best value obtained so far of 0.61. The FNO still lags behind the FFNO but performs better in the lower region on the jet offline datasets compared to the baselines. Interestingly, the FCNN trained on *ME5000* and evaluated on the jets offline dataset produces significantly worse results than the baseline. This suggests that the network might have over-learned the patterns of the eddy-driven flow. As for the U-NET, its results fall short of those of the FCNN.

Online

As depicted in Fig.25 and 26, the FCNN still struggles to capture the energy spectrum, whereas the U-NET and FFNO perform well. A nice observation is the U-NET remarkable ability to reproduce spatial distributions, visible in Fig.27 and Fig.28. However, the FFNO similarity scores remains low and rarely reach the positive range. By contrast to phase 1, from a visual perspective, the simulation dynamics of the FFNO appear to align more closely with the high-resolution simulation in phase 2, as illustrated in Fig.29 and 30.

Conclusion

To sum up, Tab.3 highlights overall improvement in metric scores. In particular, the FFNO performance stands out, giving positive values in the lower layer for jet-driven flows. Furthermore, Fig.25 and 26 reveal that the FCNN still struggles with the energy spectrum, whereas U-NET and FFNO excel. In addition to that, the U-NET effectively reproduces spatial distributions (Fig.27), while FFNO lags in similarity scores. In comparison to phase 1, phase 2 simulations for the FFNO demonstrate better alignment with high-resolution simulations (Fig.29 and 30).

	Dataset	Input	Output	Loss (Val.)	ρ_1	ρ_2	R_1^2	R_2^2
FCNN	ME5000	(q, u, v)	ϕ_q	0.03	0.99	0.98	0.97	0.96
-	-	-	-	-	0.94	0.76	-0.23	-11.8
-	MJ5000	(q, u, v)	ϕ_q	0.10	0.98	0.88	0.79	0.32
-	-	-	-	-	0.95	0.91	0.84	0.80
UNET	ME5000	(q, u, v)	ϕ_q	0.23	0.94	0.87	0.88	0.76
-	-	-	-	-	0.88	0.58	-0.40	-12.34
-	MJ5000	(q, u, v)	ϕ_q	0.50	0.81	0.66	0.59	0.21
-	-	-	-	-	0.89	0.69	0.74	0.42
FNO	ME5000	(q, u, v)	ϕ_q	0.50	0.72	0.71	0.52	0.50
-	-	-	-	-	0.66	0.27	0.42	-0.23
-	MJ5000	(q, u, v)	ϕ_q	0.88	0.48	0.36	0.13	-0.17
-	-	-	-	-	0.72	0.42	0.51	0.13
FFNO	ME5000	(q, u, v)	ϕ_q	0.19	0.91	0.90	0.83	0.80
-	-	-	-	-	0.89	0.59	0.78	0.24
-	MJ5000	(q, u, v)	ϕ_q	0.38	0.89	0.73	0.79	0.53
-	-	-	-	-	0.91	0.78	0.84	0.61
FCNN	UE5000	q	$S_{q_{\text{total}}}$	0.09	0.91	0.95	0.82	0.90
-	-	-	-	-	0.86	0.82	-0.52	-4.63
FCNN	UE5000	q	S_q	0.09	0.91	0.95	0.82	0.90
-	-	-	-	-	0.86	0.82	-0.52	-4.63
FCNN	UE5000	q	ϕ_q	0.05	0.98	0.97	0.97	0.94
-	-	-	-	-	0.95	0.76	-0.22	-8.07

Table 3: | **Offline - Phase 2** | Summarizing table of offline results. The white lines represent results for the **eddies offline** dataset, while the colored lines correspond to the **jets offline** dataset. Along with the training conditions, the validation loss (calculated on an unseen dataset of 5000 samples; for eddy training, the corresponding dataset is named **eddies validation**, and likewise for jets and the full dataset) is shown. The metric values for R^2 (see Eq.25) and ρ (see Eq.26) are given for both the upper layer (i = 1) and lower layer (i = 2).

PHASE 3

In this third phase, neural networks are trained on datasets with more samples from different simulations of the same flow type (*ME10000*, *ME20000*, *MJ10000* and *MJ20000*). The idea is to determine if a substantial increase in data volume improves the learned parameterizations quality.

Offline

From Tab.4, it is evident that increasing the total number of samples in each training dataset leads to improvement in almost every metric. It is worth noting that neural networks operating in the spatio-temporal domain, like FCNN and UNET, obtained worse results when trained on *ME20000* and evaluated on jets offline. This might be because these networks overly focus on eddy patterns, making them struggle to generalize to jet flows. However, the reverse is not true. The fact that, by contrast to the neural networks operating in the space-time domain, both Fourier Neural Operators show improvement emphasizes the power of performing computations in the spectral domain.

Online

The power spectrum, as depicted in Fig.31 and 32, is still well reproduced by the U-NET and FFNO, but not by the FCNN and FNO. For the first time, almost every similarity metric of the FFNO exceeds the threshold of 0.5 set to filter out poor results from the plots. This indicates an improvement, even though it still struggles to replicate the distribution of the high-resolution simulation.

From a visual point of view, both Fourier Neural Operators produce more aesthetically pleasing results that seems to be in line with the high resolution simulation compared to the baseline parameterizations, as seen in Fig.35 and 36.

Conclusion

Increasing sample numbers improves metrics (Tab.4). However, FCNN and UNET struggle when trained on *ME20000* and tested on jets, possibly due to over-learning eddy patterns. Nevertheless, Fourier Neural Operators improve with an increase in spectral domain representation efficiency. In the online tests, the U-NET and FFNO keep exceling in power spectrum reproduction (Fig.31 and 32). The FFNO metrics have improved by exceeding our aribtrary 0.5 threshold, though distribution replication remains a challenge. Both Fourier Neural Operators yield visually appealing results, surpassing baseline parameterizations (Fig.35 and 36).

	Dataset	Input	Output	Loss (Val.)	ρ_1	ρ_2	R_1^2	R_2^2
FCNN	ME20000	(q, u, v)	ϕ_q	0.02	0.99	0.99	0.98	0.97
-	-	-	-	-	0.94	0.73	-0.34	-15.23
-	MJ20000	(q, u, v)	ϕ_q	0.07	0.98	0.81	0.75	0.26
-	-	-	-	-	0.95	0.93	0.88	0.85
UNET	ME20000	(q, u, v)	ϕ_q	0.19	0.95	0.89	0.9	0.8
-	-	-	-	-	0.89	0.65	-0.43	-14.45
-	MJ20000	(q, u, v)	ϕ_q	0.40	0.87	0.69	0.62	0.21
-	-	-	-	-	0.90	0.74	0.80	0.54
FNO	ME20000	(q, u, v)	ϕ_q	0.41	0.78	0.77	0.6	0.59
-	-	-	-	-	0.68	0.28	0.41	-0.34
-	MJ20000	(q, u, v)	ϕ_q	0.71	0.57	0.39	0.28	-0.14
-	-	-	-	-	0.79	0.54	0.62	0.28
FFNO	ME20000	(q, u, v)	ϕ_q	0.15	0.93	0.92	0.87	0.85
-	-	-	-	-	0.93	0.66	0.86	0.37
-	MJ20000	(q, u, v)	ϕ_q	0.30	0.93	0.79	0.86	0.62
-	-	-	-	-	0.94	0.83	0.88	0.7
FCNN	UE5000	q	$S_{q_{\text{total}}}$	0.09	0.91	0.95	0.82	0.90
-	-	-	-	-	0.86	0.82	-0.52	-4.63
FCNN	UE5000	q	S_q	0.09	0.91	0.95	0.82	0.90
-	-	-	-	-	0.86	0.82	-0.52	-4.63
FCNN	UE5000	q	ϕ_q	0.05	0.98	0.97	0.97	0.94
-	-	-	-	-	0.95	0.76	-0.22	-8.07

Table 4: | **Offline - Phase 3** | Summarizing table of offline results. The white lines represent results for the **eddies offline** dataset, while the colored lines correspond to the **jets offline** dataset. Along with the training conditions, the validation loss (calculated on an unseen dataset of 5000 samples; for eddy training, the corresponding dataset is named **eddies validation**, and likewise for jets and the full dataset) is shown. The metric values for R^2 (see Eq.25) and ρ (see Eq.26) are given for both the upper layer (i = 1) and lower layer (i = 2).

PHASE 4

In this fourth phase, neural networks are trained on datasets containing samples from both flow types. Besides studying the impact of shifting to a full dataset of 5000 samples (*F5000*), we also assess the effects of training on a much larger dataset of 40000 samples (*F40000*) at the same time.

Offline

From the Tab.5, it is evident that this phase marks the most significant shift in results. The Fourier Neural Operators have both improved their metric scores when compared to the former phases. However, for the FCNN and UNET, their negative scores in the lower layer of the jets online simulation still persist. Notably, the FFNO results trained on *F40000* are impressive. Although the metrics for the eddies online simulation are slightly lower than the baselines, the results for the jets simulation greatly outperform the baselines in terms of R^2 but are still a behind for ρ.

Online

The same shift in results is observable in Fig.37 concerning the power spectra. As previously discussed, strong offline performance does not guarantee smooth online simulation. Despite successfully completing both eddy and jet online simulations without violating the CFL number, the FFNO energy spectrum spikes in both cases. In additiont to that the FCNN still gives poor results, the U-NET worsens, but surprisingly, the FNO yields acceptable results. Additionally, all similarity scores (see Fig.38) for all parameterizations have significantly dropped. Nonetheless, visualizations suggest that parameterizations can produce physics-like results, except for the U-NET, which notably fails to distribute energy correctly in its simulation, as shown in Fig.39.

Conclusion

Despite notable offline improvements for the FFNO due to the use of full datasets, the online results remained poor. In order to overcome this problem, we will investiguate further the FFNO architecture. Indeed, the solution might be found by optimizing the training conditions as well as the architecture used. Thus, the focus of the folllowing phases will on be toward the FFNO, keeping potential vorticity and velocity field as inputs, along with the conservative subgrid scale processes formulation for output, i.e. $\mathbf{\Phi}_q$.

	Dataset	Input	Output	Loss (Val.)	ρ_1	ρ_2	R_1^2	R_2^2
FCNN	F5000	(q, u, v)	ϕ_q	0.04	0.98	0.97	0.95	0.94
-	-	-	-	-	0.94	0.85	0.40	-5.75
-	F40000	(q, u, v)	ϕ_q	0.02	0.99	0.98	0.97	0.93
-	-	-	-	-	0.95	0.85	0.41	-4.67
UNET	F5000	(q, u, v)	ϕ_q	0.25	0.93	0.84	0.84	0.68
-	-	-	-	-	0.89	0.63	0.32	-4.91
-	F40000	(q, u, v)	ϕ_q	0.20	0.94	0.88	0.87	0.75
-	-	-	-	-	0.90	0.69	0.41	-6.11
FNO	F5000	(q, u, v)	ϕ_q	0.54	0.71	0.69	0.5	0.47
-	-	-	-	-	0.70	0.37	0.49	0.13
-	F40000	(q, u, v)	ϕ_q	0.41	0.8	0.77	0.64	0.60
-	-	-	-	-	0.8	0.47	0.63	0.22
FFNO	F5000	(q, u, v)	ϕ_q	0.21	0.91	0.89	0.84	0.79
-	-	-	-	-	0.91	0.70	0.83	0.49
-	F40000	(q, u, v)	ϕ_q	0.15	0.94	0.93	0.88	0.86
-	-	-	-	-	0.94	0.79	0.89	0.62
FCNN	UE5000	q	$S_{q_{\text{total}}}$	0.09	0.91	0.95	0.82	0.90
-	-	-	-	-	0.86	0.82	-0.52	-4.63
FCNN	UE5000	q	S_q	0.09	0.91	0.95	0.82	0.90
-	-	-	-	-	0.86	0.82	-0.52	-4.63
FCNN	UE5000	q	ϕ_q	0.05	0.98	0.97	0.97	0.94
-	-	-	-	-	0.95	0.76	-0.22	-8.07

Table 5: | **Offline - Phase 4** | Summarizing table of offline results. The white lines represent results for the **eddies offline** dataset, while the colored lines correspond to the **jets offline** dataset. Along with the training conditions, the validation loss (calculated on an unseen dataset of 5000 samples; for eddy training, the corresponding dataset is named **eddies validation**, and likewise for jets and the full dataset) is shown. The metric values for R^2 (see Eq.25) and ρ (see Eq.26) are given for both the upper layer (i = 1) and lower layer (i = 2).

PHASE 5

In this fifth phase, we focus on investigating the training conditions of the Factorized Fourier Neural Operators (FFNO). Originally, the training condition were the one of Ross et al., 2023. However, the objective now is to determine whether the FFNO can perform better, especially in online testing, with different combinations of optimizer (ADAM or SGD), scheduler (constant, cosine warmup, cosine, cyclic, exponential, and multi-step), and learning rate ($\gamma = 0.1, 0.01, 0.001$). Finally, the training was done using the *F5000* dataset.

Offline

All the results are depicted in Fig.40 and summarized in Tab.6. The first row in the table corresponds to the training conditions used in Ross et al., 2023, the second row shows the conditions recommended in the original FFNO paper by Tran et al., 2021, and the last rows display two of our results. As it can be seen, training with a cosine scheduler enhances the results, but a constant learning rate of 0.01 yields even better results. The most favorables are achieved with a learning rate of 0.01. Nevertheless, it is worth noting that when $\gamma = 0.001$, the best results also arises from using a constant scheduler. Finally, a small improvement in metric values is easily observed between the original training conditions and the best ones obtained.

Online

The online testing results underscore the significance of benchmarking the parameterization in a real-case simulation. As depicted in Fig.41, the FFNO, which exhibited superior offline results using a constant learning rate of 0.01, now demonstrates an energy spectrum that explodes during online testing. As a consequence, the second-best results take precedence as the new best results. This implies that the FFNO will be trained using a constant scheduler rather than a multi-step or even cosine scheduler.

Conclusion

The Factorized Fourier Neural Operator will now be trained using the ADAM optimizer and a constant learning rate of 0.001 since it has improved all its metric scores. Consequently, the final step of this study involves exploring the architecture of the FFNO while using these newly identified training conditions to obtain the final parameterization of this study.

	Opti.	Sched.	Learn. Rate	Loss (Val.)	ρ_1	ρ_2	R_1^2	R_2^2
FFNO	ADAM	Multi-Step	0.001	0.22	0.91	0.89	0.84	0.79
-	-	-	-	-	0.91	0.69	0.83	0.47
FFNO	ADAM	Cosine	0.001	0.20	0.92	0.90	0.84	0.80
-	-	-	-	-	0.92	0.71	0.84	0.5
FFNO	ADAM	Constant	0.01	0.17	0.93	0.91	0.87	0.83
-	-	-	-	-	0.93	0.76	0.87	0.58
FFNO	ADAM	Constant	0.001	0.19	0.92	0.90	0.85	0.81
-	-	-	-	-	0.92	0.73	0.85	0.53
FCNN	UE5000	q	$S_{q_{\text{total}}}$	0.09	0.91	0.95	0.82	0.90
-	-	-	-	-	0.86	0.82	-0.52	-4.63
FCNN	UE5000	q	S_q	0.09	0.91	0.95	0.82	0.90
-	-	-	-	-	0.86	0.82	-0.52	-4.63
FCNN	UE5000	q	ϕ_q	0.05	0.98	0.97	0.97	0.94
-	-	-	-	-	0.95	0.76	-0.22	-8.07

Table 6: | **Offline - Phase 5** | Summarizing table of offline results. The white lines represent results for the **eddies offline** dataset, while the colored lines correspond to the **jets offline** dataset. The inputs of the tested parameteriations are (q, u, v) and the output is the subgrid flux $\mathbf{\Phi}_q$. Along with the different training conditions explored, the validation loss (calculated on an unseen dataset of 5000 samples; for eddy training, the corresponding dataset is named **eddies validation**, and likewise for jets and the full dataset) is shown. The metric values for R^2 (see Eq.25) and ρ (see Eq.26) are given for both the upper layer (i = 1) and lower layer (i = 2).

PHASE 6

In this sixth phase, the focus is on diving deeper into the architecture of the FFNO. One of the main advantages of this architecture, compared to the FNO, lies in its ability to scale to more than the 4 Fourier layers allowed originally by the FNO. Additionally, it is interesting to investigate the role of the spectral modes for prediction. To conduct a more comprehensive analysis, modifications have been made to the FFNO library, enabling the use of a pass-band filter instead of a simple low-pass. This means specific modes within a range can now be chosen for prediction, as opposed to relying solely on the maximum mode and all modes up to it. Moreover, examining the influence of the latent space representation size, used for enhancing the original input, is important. In conclusion, this study will explore the impacts of number of layers, latent space size (= width of Fourier layer), the modes selected and the training was done using *F5000* and batch size of 32.

Offline

All results are depicted in Fig.44 and 45, with a summary presented in Tab.7. The optimal configuration emerges with a width of 128. Indeed, an important difference in metric values is observed between modes $(0, 8)$ and configurations using more modes, while distinctions between $(16, 24)$ and $(24, 32)$ modes are less pronounced. Interestingly, the performance gap between a width of 128 using the initial 32 modes with 24 layers diminishes when compared to the setup employing 4 layers. Introduction of a pass band filter underscores the significance of the first 8 modes, yielding superior results for the parameterization by contrast to using any other 8 modes. It is important to highlight that, the FFNO demonstrates exceptional performance with a width of 128, 24 layers, and the first 32 modes, significantly outperforming the baselines even in the case using only 4 layers.

Online

Each configuration exhibits stable energy spectra that aligns with either the low-resolution or high-resolution spectra (see Fig.46). Notably, the FFNO, in its optimal configuration, presents improved similarity scores in the spectral domain (see Fig.47). However, this improvement is not yet mirrored in the spatial distribution of the flow quantities of interest.

Conclusion

In conclusion, the best FFNO configuration uses a width of 128, 24 layers, and the first 32 modes consistently outperforms baselines across various metrics. It exhibits stable energy spectra and improved similarity scores in the spectral domain. However, there is still room for improvements regarding the spatial distribution of flow quantities.

	Width	Modes	Num. Layers	Loss (Val.)	ρ_1	ρ_2	R_1^2	R_2^2
FFNO	128	(0, 8)	24	0.41	0.72	0.77	0.51	0.59
-	-	-	-	-	0.75	0.47	0.55	0.21
FFNO	128	(0, 16)	24	0.09	0.96	0.96	0.93	0.92
-	-	-	-	-	0.97	0.85	0.94	0.72
FFNO	128	(0, 24)	24	0.05	0.99	0.98	0.97	0.96
-	-	-	-	-	0.99	0.93	0.98	0.85
FFNO	128	(0, 32)	24	0.01	0.99	0.99	0.98	0.98
-	-	-	-	-	0.99	0.96	0.99	0.93
FFNO	128	(8, 16)	24	0.42	0.72	0.77	0.52	0.6
-	-	-	-	-	0.67	0.33	0.45	0.10
FFNO	128	(16, 24)	24	0.70	0.54	0.57	0.28	0.30
-	-	-	-	-	0.47	0.22	0.23	0.05
FFNO	128	(24, 32)	24	0.96	0.19	0.22	0.03	0.03
-	-	-	-	-	0.24	0.19	0.10	0.02
FFNO	128	(0, 32)	4	0.04	0.99	0.98	0.97	0.96
-	-	-	-	-	0.99	0.93	0.98	0.86
FCNN	UE5000	q	$S_{q_{\text{total}}}$	0.09	0.91	0.95	0.82	0.90
-	-	-	-	-	0.86	0.82	-0.52	-4.63
FCNN	UE5000	q	S_q	0.09	0.91	0.95	0.82	0.90
-	-	-	-	-	0.86	0.82	-0.52	-4.63
FCNN	UE5000	q	ϕ_q	0.05	0.98	0.97	0.97	0.94
-	-	-	-	-	0.95	0.76	-0.22	-8.07

Table 7: | **Offline - Phase 6** | Summarizing table of offline results. The white lines represent results for the **eddies offline** dataset, while the colored lines correspond to the **jets offline** dataset. The inputs of the tested parameteriations are (q, u, v) and the output is the subgrid flux $\mathbf{\Phi}_q$. Along with the different configurations explored, the validation loss (calculated on an unseen dataset of 5000 samples; for eddy training, the corresponding dataset is named **eddies validation**, and likewise for jets and the full dataset) is shown. The metric values for R^2 (see Eq.25) and ρ (see Eq.26) are given for both the upper layer (i = 1) and lower layer (i = 2).

Chapter Six

The end of an adventure

In this final chapter, a summary of the entire work will be provided, followed by a discussion where notable observations and potential improvements will be highlighted. This will containt insights gathered during the journey, including ideas for further exploration left for the reader to consider.

6.1 CONCLUSION

The journey undertaken throughout this work, as complex and fascinating as it is, can be summarized as follows: Over the past decades, global climate change has continuously led to increased natural disasters, humanitarian crises, and ocean oxygen deoxygenation. To address these challenges, it is important, as engineers, to find a model that enables the simulation, understanding, and prediction of the Earth climate system. However, working at the Earth scale is time-consuming, and the most viable solution is to lower the simulation resolution. Yet, this approach results in missing some physics described by the equations, leading to less accurate representation of the intended dynamics.

A simple solution to address this problem is obtained by considering these missing processes, often termed subgrid scale processes, through parameterization. Analytically, this issue, also known as turbulence closure modeling, has been present for a century. Despite some promising ideas, they all share a common limitation: they have to increase artificially some physical properties of the fluid whose motion we aim to describe. As a result, the dynamics depicted in the simulation do not truly match the actual behavior. In response to the rapidly growing field of machine learning, scientists have spent the past decade exploring its intersection with fluid mechanics. This idea involves using a neural network to serve as the parameterization for the missing contributions from subgrid scale processes.

This master thesis is centered around the study by Ross et al., 2023. Over the years, they developed varying parameterizations, each with its advantages and disadvantages. To more effectively and rigorously evaluate their quality, they established a benchmarking framework, thoroughly described in their paper. Thus, the goal of our study was rather simple, even if the physical context is not: Is it possible to find a new parameterization for the subgrid scale processes that works better (this being assessed with their framework) than the one they presented? To answer this problem, the chosen path was found as follows: in their future work section, they explained that, even if they have investigated a lot of different ideas, one thing they did not do is study the impact of more complex datasets.

Therefore, this will be our chosen path to drive this study. Furthermore, instead of merely using the fully-convolutional neural network they used multiple times and attempting to enhance it, we have also decided to incorporate a U-NET, a state of the art model in image recognition. But most importantly of all, we have also ventured into exploring the use of the Fourier Neural Operators for subgrid scale processes parameterization, a path that, as for today, has not been taken until now. After carefully explaining the physics to understand computational fluid dynamics, ocean modeling, and the mathematical formulation of the missing contributions we aim to compute, we proceeded with an in-depth description of how these new Fourier Neural Operators operate. Subsequently, we covered the datasets used, the training setup, and the metrics. Throughout our analysis, we were able to highlight several key factors:

- **FCNN**: The architecture displayed some improvement in its predictive capabilities when using our more complex datasets. Nevertheless, using this neural network for simulation still appears unfavorable due to inadequate energy spectra representation. This final results suggest that the architecture should be simply revisited if one hopes to finally obtain a more coherent online simulation.

- **U-NET**: This state of the art architecture did not outmatch the FCNN in any test regarding the predicting capabilities. However, the key observation that can be made is its ability to correctly redistribute spatially flow quantities. In other words, it seems like this architecture correlates the longest with the dynamic of the high resolution simulation. Thus, an idea would be to create an architecture that uses the best of both.

- **FNO** and **FFNO**: This study showed the power of Fourier Neural Operators, which, in contrast to everything done so far in this situation, perform a major part of their computation in the spectral domain. At the end of this study, we obtained a final parameterization for the subgrid scale processes that outperform the prediction capabilities of the networks presented in their papers Ross et al., 2023. Moreover, when using metrics of their framework, we also observe that the energy deficiency of the simulation was more accurately corrected. However, as for many engineering problems, nothing comes without a tradeoff. While the recently acquired parameterization shows highly promising prediction results, its drawback emerges from the inability to precisely align the spatial corrections with the high-resolution simulation dynamics. As a result, the simulation's behavior deviates from that of the high-resolution.

6.2 DISCUSSION AND FUTURE WORK

Even though this work was complex due to the fascinating but complicated intersection of physics and deep learning, I do think that we managed to extract some valuable information regarding this problem of subgrid scale processes parameterization. Throughout this work and also after completing it, I realized that some choices made could have been better, and also, I had ideas about what could be explored to try to go further. This reflection path can be broken down as follows:

DATASETS

- While generating datasets, I noticed that from time to time, the bottom layer of a jet-driven flow looked a lot like what can be found in an eddy-driven flow. This could be the reason why a neural network on an eddy-driven flow performs worse on the offline test made on jets than the other way around. Therefore, it might have been interesting to change the proportions of samples from the upper and lower layers of a jet-driven dataset to something like 75-25 instead of a mere 50-50.

- In the case of a jet-driven flow, the sampling can only start after 6 years; otherwise, the quasi-steady state solution is not reached. Therefore, since the simulation is still performed on a 10-year interval and the same number of samples is asked, it might be possible that the sampling of the simulation is poorly performed, leading to samples that are temporarily speaking too close to one another and thus the dynamics of the flow are not explored well enough. Thus, a solution could be to run the simulation for a bit longer to put more space between sampling moments.

NEURAL NETWORKS

- In the Factorized Fourier Neural Operator architecture, I wonder what would happen if the modulation of the frequency spectrum between the input and the target output was performed with something more advanced than a simple weight matrix. Indeed, what would happen if the modulation was achieved using something capable of creating non-linear relationships, such as a fully connected neural network?

- As we have observed, the best results obtained were using the first 32 modes of the frequency spectrum. Instead of using a single weight matrix for all modes simultaneously, it could be interesting to have Fourier layers that specialize in modulating different parts of the spectrum, and then recombine their results. This approach could be very intriguing. Imagine successfully creating a parameterization using deep learning models based on Fourier analysis. Each of these Fourier blocks would focus on learning how to modulate a part of the spectrum, and at the end of the network, a final layer would mix the results. Using a technique like the one described in Cranmer et al., 2020, we could attempt to extract the learned relationships and find parameterizations for the missing subgrid scale contributions occurring at different scales of our problem. Finally, we would learn how to combine these equations to create a complete one. This could provide an interpretable parameterization that would offer further insights into how to address this long-standing problem.

- The main issue with the Fourier Neural Operators is their inability to reproduce spatial distributions of flow quantities the same way a high-resolution simulation does, i.e. the corrected low-resolution simulation diverges too quickly from the dynamics of the high-resolution. Another idea that I tried to investigate, unfortunately without success, is to create a new architecture that would combine the benefits of computing operations in the spectral domain (better energy correction) and performing operations in the spatiotemporal domain (similar to the U-NET and FCNN), allowing for a more accurate spatial representation of the flow variables. This to me, is one of the key ideas that have emerged from this work.

- Even if the results of the FFNO are impressive, it could have been interesting to evaluate its performance using an early stopped version of the network during training. Indeed, it seems rather obvious that the network might have overfitted the data a bit during training for 50 epochs. Fortunately, all the neural network states from every epoch are saved, but time is the only thing missing to investigate what would effectively happen.

- A metric that I did not use but which is really important is the decorrelation time. It is a measure of the time it takes for the low-resolution simulation to diverge from the high resolution. In their framework, the way online metrics are computed, if not averaged over time, is at the end of the simulation using the last time step. Therefore, by checking the decorrelation time of the simulation, it could be interesting to recompute these metrics at a moment (if it exists) where the simulations still match one another (even if that might not be the case in this context).

- Even if the FFNO gave promising results, I am a bit disappointed with the time it takes to run a simulation. Indeed, another issue with this architecture is the fact that the Fourier transform and its inverse must be computed at each Fourier Layer, which is a pretty time-consuming operation. In addition to that, both the original input and the one being transformed in the spectral domain must be stored on the GPU, which leads to a huge memory need during training. During Phase 6, it was not possible to train the 24-layer FFNO without using 4 GPUs.

OTHERS

Although this benchmarking framework is a significant leap forward in the field of subgrid processes parameterizations as it allows a rigorous assessment of their quality, I still think that it lacks a bit of simplicity. Due to the already substantial complexity of the problem studied, it is not possible to use any functions without wondering what is given as input and what is the physical meaning of the output. In addition to that, making sense of the results is still a challenging task due to the complexity of these metrics and the fact that their interpretability makes sense only if the reader has a minimum background knowledge in both fluid mechanics and statistics. Therefore, a large portion of my time was dedicated to making the framework more user-friendly by creating a notebook that would allow me to use these tools more simply. For anyone curious about how to use PyQG or how to use their framework, I highly encourage you to try my master thesis notebook available on my GitHub page. I tried to document as much as possible what is possible to do, how to generate data, train a parameterization, evaluate it both offline and online as simply as possible.

References

Ross, Andrew et al. (2023). "Benchmarking of machine learning ocean subgrid parameterizations in an idealized model". In: *Journal of Advances in Modeling Earth Systems* 15.1, e2022MS003258.

Li, Zongyi et al. (2020). "Fourier neural operator for parametric partial differential equations". In: *arXiv preprint arXiv:2010.08895*.

Breitburg, Denise et al. (2018). "Declining oxygen in the global ocean and coastal waters". In: *Science* 359.6371, eaam7240.

Karstensen, Johannes, Lothar Stramma, and Martin Visbeck (2008). "Oxygen minimum zones in the eastern tropical Atlantic and Pacific oceans". In: *Progress in Oceanography* 77.4, pp. 331–350.

Capet, Arthur et al. (2016). "Decline of the Black Sea oxygen inventory". In: *Biogeosciences* 13.4, pp. 1287–1297.

Stanev, EV et al. (2013). "Oxygen dynamics in the Black Sea as seen by Argo profiling floats". In: *Geophysical Research Letters* 40.12, pp. 3085–3090.

Kolmogorov, Andrei Nikolaevich (1941). "The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers". In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 434.1890, pp. 9–13.

Lévy, Marina, Laure Resplandy, et al. (2012). "Grid degradation of submesoscale resolving ocean models: Benefits for offline passive tracer transport". In: *Ocean Modelling* 48, pp. 1–9.

Resplandy, Laure, Marina Lévy, Gurvan Madec, et al. (2011). "Contribution of mesoscale processes to nutrient budgets in the Arabian Sea". In: *Journal of Geophysical Research: Oceans* 116.C11.

Lévy, Marina, Raffaele Ferrari, et al. (2012). "Bringing physics to life at the submesoscale". In: *Geophysical Research Letters* 39.14.

Martin, Adrian P et al. (2015). "An observational assessment of the influence of mesoscale and submesoscale heterogeneity on ocean biogeochemical reactions". In: *Global Biogeochemical Cycles* 29.9, pp. 1421–1438.

Ramachandran, Sanjiv, Amit Tandon, and Amala Mahadevan (2014). "Enhancement in vertical fluxes at a front by mesoscale-submesoscale coupling". In: *Journal of Geophysical Research: Oceans* 119.12, pp. 8495–8511.

Resplandy, Laure, Marina Lévy, Francesco d'Ovidio, et al. (2009). "Impact of submesoscale variability in estimating the air-sea CO2 exchange: Results from a model study of the POMME experiment". In: *Global Biogeochemical Cycles* 23.1.

Djath, Bughsin' et al. (2014). "Multiscale dynamical analysis of a high-resolution numerical model simulation of the Solomon Sea circulation". In: *Journal of Geophysical Research: Oceans* 119.9, pp. 6286–6304.

Su, Zhan et al. (2018). "Ocean submesoscales as a key component of the global heat budget". In: *Nature communications* 9.1, p. 775.

Lévy, Marina and Adrian P Martin (2013). "The influence of mesoscale and submesoscale heterogeneity on ocean biogeochemical reactions". In: *Global Biogeochemical Cycles* 27.4, pp. 1139–1150.

Prandtl, Ludwig (1925). "7. Bericht über Untersuchungen zur ausgebildeten Turbulenz". In: *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik* 5.2, pp. 136–139.

Smagorinsky, Joseph (1963). "General circulation experiments with the primitive equations: I. The basic experiment". In: *Monthly weather review* 91.3, pp. 99–164.

Singhal, AK and DB Spalding (1981). "Predictions of two-dimensional boundary layers with the aid of the k-epsilon model of turbulence". In: *Computer Methods in Applied Mechanics and Engineering* 25.3, pp. 365–383.

Menter, Florian R (1992). *Improved two-equation k-omega turbulence models for aerodynamic flows*. Tech. rep.

Jansen, Malte F and Isaac M Held (2014). "Parameterizing subgrid-scale eddy effects using energetically consistent backscatter". In: *Ocean Modelling* 80, pp. 36–48.

Jansen, Malte F, Isaac M Held, et al. (2015). "Energy budget-based backscatter in an eddy permitting primitive equation model". In: *Ocean Modelling* 94, pp. 15–26.

Pope, Stephen B (1975). "A more general effective-viscosity hypothesis". In: *Journal of Fluid Mechanics* 72.2, pp. 331–340.

Ling, Julia, Andrew Kurzawski, and Jeremy Templeton (2016). "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance". In: *Journal of Fluid Mechanics* 807, pp. 155–166.

Sharma, Pushan et al. (2023). "A Review of Physics-Informed Machine Learning in Fluid Mechanics". In: *Energies* 16.5, p. 2343.

Bolton, Thomas and Laure Zanna (2019). "Applications of deep learning to ocean data inference and subgrid parameterization". In: *Journal of Advances in Modeling Earth Systems* 11.1, pp. 376–399.

Tran, Alasdair et al. (2021). "Factorized fourier neural operators". In: *arXiv preprint arXiv:2111.13802*.

Bonev, Boris et al. (2023). "Spherical Fourier Neural Operators: Learning Stable Dynamics on the Sphere". In: *arXiv preprint arXiv:2306.03838*.

Cranmer, Miles et al. (2020). "Discovering symbolic models from deep learning with inductive biases". In: *Advances in Neural Information Processing Systems* 33, pp. 17429–17442.

Quattromini, Michele et al. (2023). "Operator learning of RANS equations: a Graph Neural Network closure model". In: *arXiv preprint arXiv:2303.03806*.

Zanna, Laure and Thomas Bolton (2020). "Data-driven equation discovery of ocean mesoscale closures". In: *Geophysical Research Letters* 47.17, e2020GL088376.

Griffies, Stephen (2018). *Fundamentals of ocean climate models*. Princeton university press.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, pp. 234–241.

Çiçek, Özgün et al. (2016). "3D U-Net: learning dense volumetric segmentation from sparse annotation". In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*. Springer, pp. 424–432.

Charney, Jule G (1971). "Geostrophic turbulence". In: *Journal of the Atmospheric Sciences* 28.6, pp. 1087–1095.

Berrone, S and D Oberto (2022). "An invariances-preserving vector basis neural network for the closure of Reynolds-averaged Navier–Stokes equations by the divergence of the Reynolds stress tensor". In: *Physics of Fluids* 34.9.

Terrapon, Vincent (2023). "Introduction to computational fluid dynamics". In: First lecture.

Kraichnan, Robert H (1976). "Eddy viscosity in two and three dimensions". In: *Journal of Atmospheric Sciences* 33.8, pp. 1521–1536.

Appendix

Even more results

Torture the data, and it will confess to anything."

- Ronald Coase

1.1 INTRODUCTION

In this appendix, you will find the offline results of phases 5 and 6, as well as all the online results involving parameterizations trained on jet-driven flows or evaluated on online simulations of jet flows. Due to the substantial number of results generated, it should be noted that all the missing offline results and online results that involve eddy-trained parameterizations or testing with eddy datasets are available in my GitHub repository. Indeed, less than a third of the results are shown in this master thesis.

Throughout this work, ensuring the reproducibility of my results was my top priority. Therefore, all the codes and notebooks created for this work or based on the benchmarking framework of Ross et al., 2023 have been thoroughly documented. This documentation not only makes it easier for those interested to understand how to use the benchmarking framework, but also allows for a comprehensive overview of all the work done. One of my proudest achievements in terms of code is the master thesis notebook, which is concise, well documented, and enables easy dataset generation, training of new parameterizations, and both offline and online testing.

https://github.com/VikVador/Ocean-subgrid-parameterizations-in-an-idealized-model-using-machine-learning

PHASE I

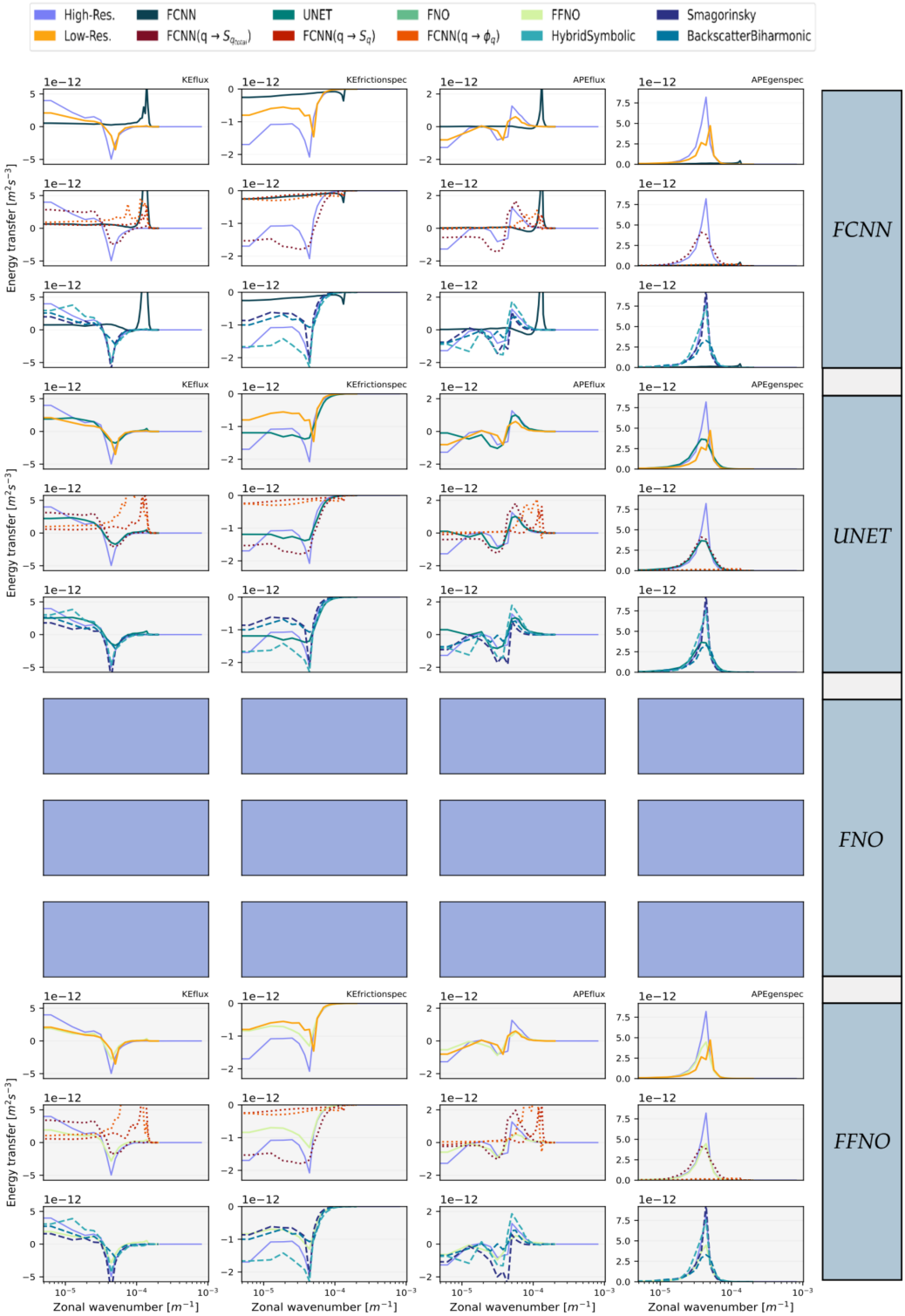


Figure 19: | **Online - Phase 1 - Energy budget** | This table displays energy spectra for **KEflux**, **KEfrictionspec**, **APEflux**, and **APEgenspec** using parameterizations grouped in Tab.2, these were trained on *UJ5000* and tested on **eddies online**. Each parameterization spectrum is compared against high-resolution and various low-resolution simulations, including neural networks from Ross et al., 2023 and analytical parameterizations from Smagorinsky, 1963; Jansen and Held, 2014; Zanna and Bolton, 2020.

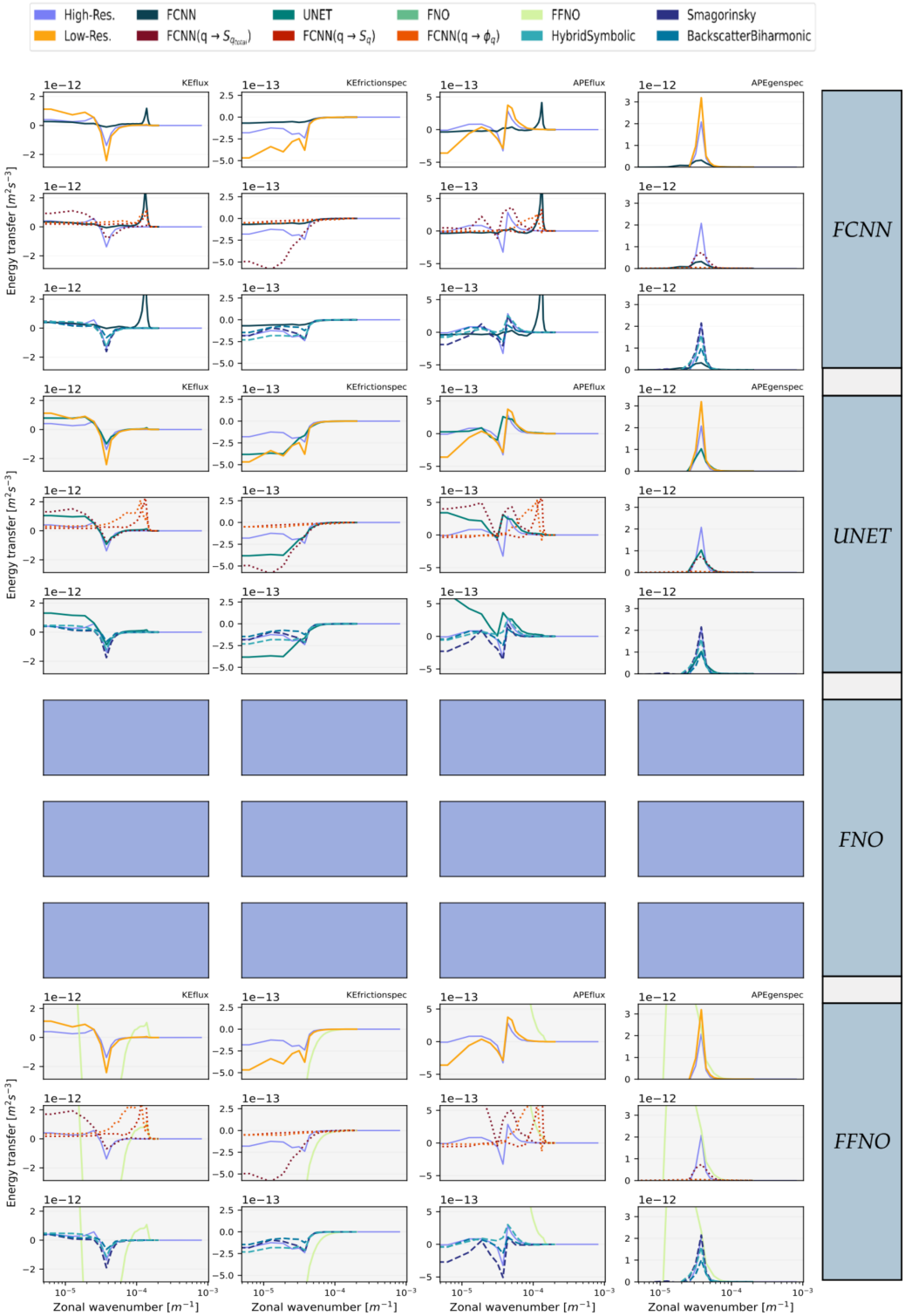


Figure 20: | **Online - Phase 1 - Energy budget** | This table displays energy spectra for **KEflux**, **KEfrictionspec**, **APEflux**, and **APEgenspec** using parameterizations grouped in Tab.2, these were trained on *UJ5000* and tested on **jets online**. Each parameterization spectrum is compared against high-resolution and various low-resolution simulations, including neural networks from Ross et al., 2023 and analytical parameterizations from Smagorinsky, 1963; Jansen and Held, 2014; Zanna and Bolton, 2020.

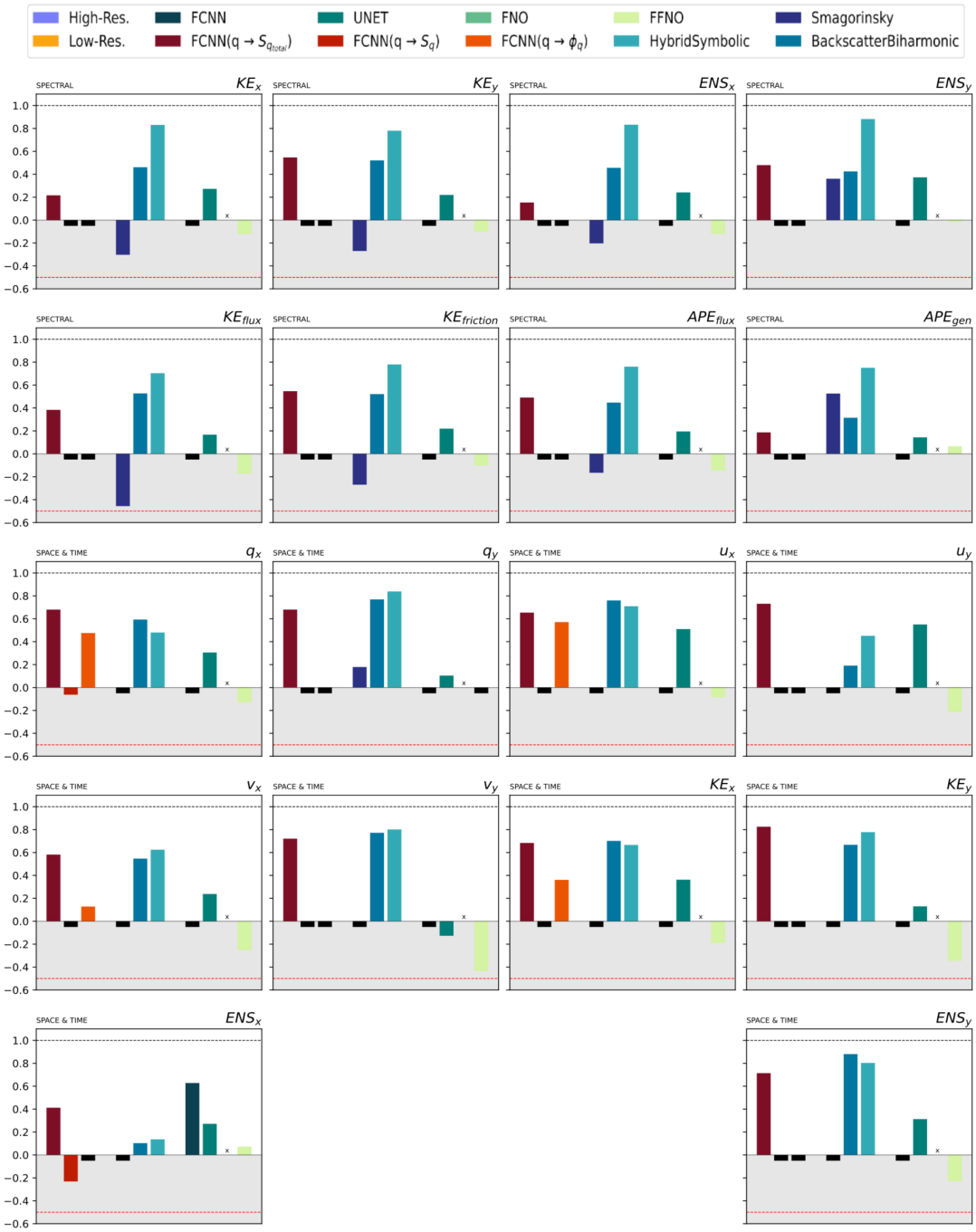


Figure 21: | **Online - Phase 1 - Similarities** | This table provides a summary of the Earth mover's distance, reformulated as a similarity metric for various flow quantities represented in either spectral or spatiotemporal domains. A value approaching 1 indicates strong agreement between the distribution obtained from high-resolution simulations and the current observations. Negative values are considered unfavorable, and values lower than -0.5 are disregarded. The tested parameterizations are grouped in Tab.2, they are trained on *UJ5000* and tested on **eddies online**. For comparison, the results of neural networks (Ross et al., 2023) and analytical parameterizations are also presented (Smagorinsky, 1963; Jansen and Held, 2014; Zanna and Bolton, 2020).

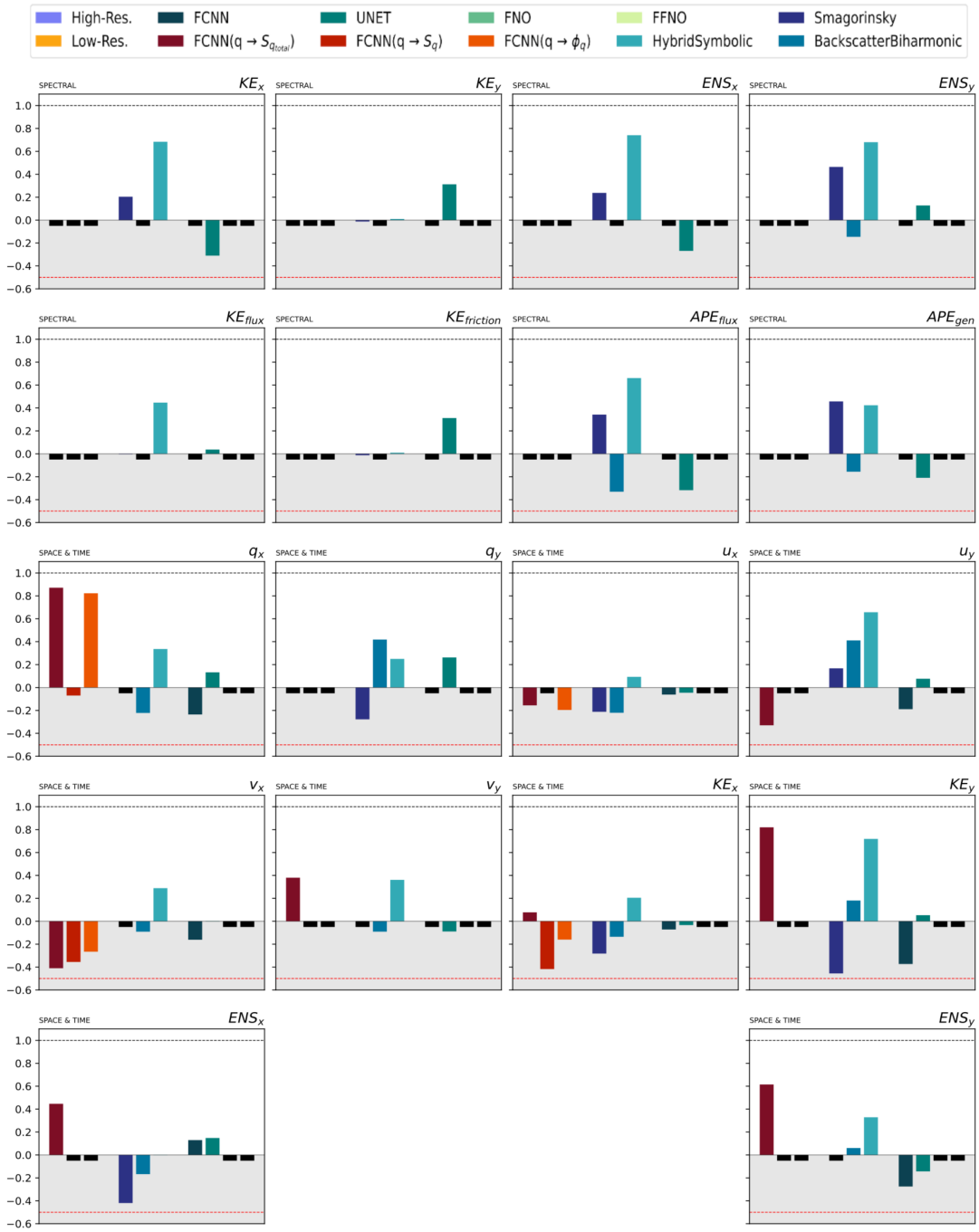


Figure 22: | **Online - Phase 1 - Similarities** | This table provides a summary of the Earth mover's distance, reformulated as a similarity metric for various flow quantities represented in either spectral or spatiotemporal domains. A value approaching 1 indicates strong agreement between the distribution obtained from high-resolution simulations and the current observations. Negative values are considered unfavorable, and values lower than -0.5 are disregarded. The tested parameterizations are grouped in Tab.2, they are trained on *UJ5000* and tested on **jets online**. For comparison, the results of neural networks (Ross et al., 2023) and analytical parameterizations are also presented (Smagorinsky, 1963; Jansen and Held, 2014; Zanna and Bolton, 2020).

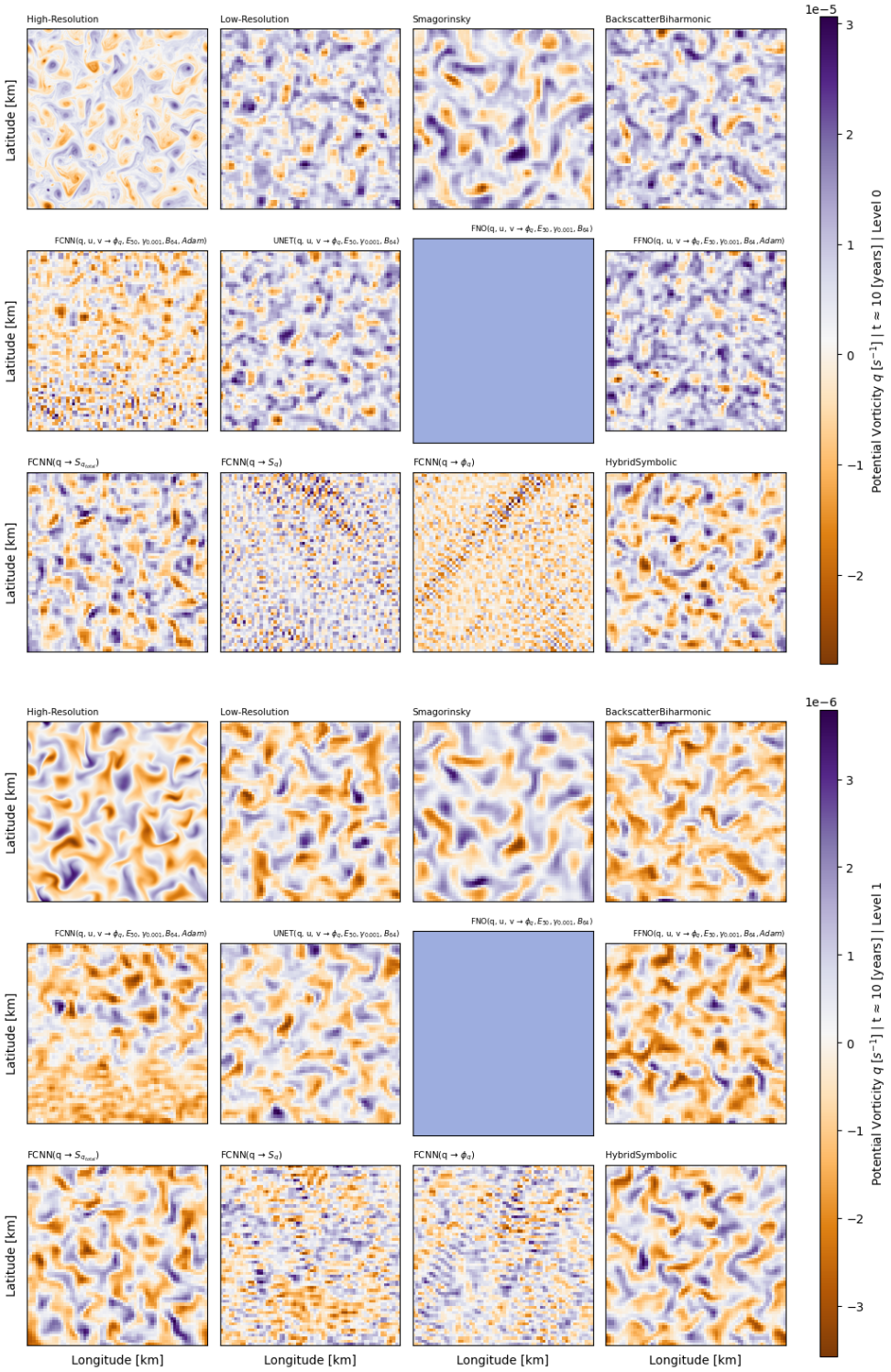


Figure 23: | **Online - Phase 1 - Potential vorticity** | Visualization of potential vorticity q for both upper (first three rows) and lower (last three rows) layers across different simulation types, indicated at the top of each image. Each image represents the q value spanning the entire computational domain after 10 years of simulations. The objective is to emphasize and visualize simulations that lose their physical relevance, becoming mere pixel grids, and to illustrate the divergence from the high-resolution simulation. Furthermore, the evaluated parameterizations are detailed in Tab.2, they were trained using *UJ5000* and tested on **eddies online**.

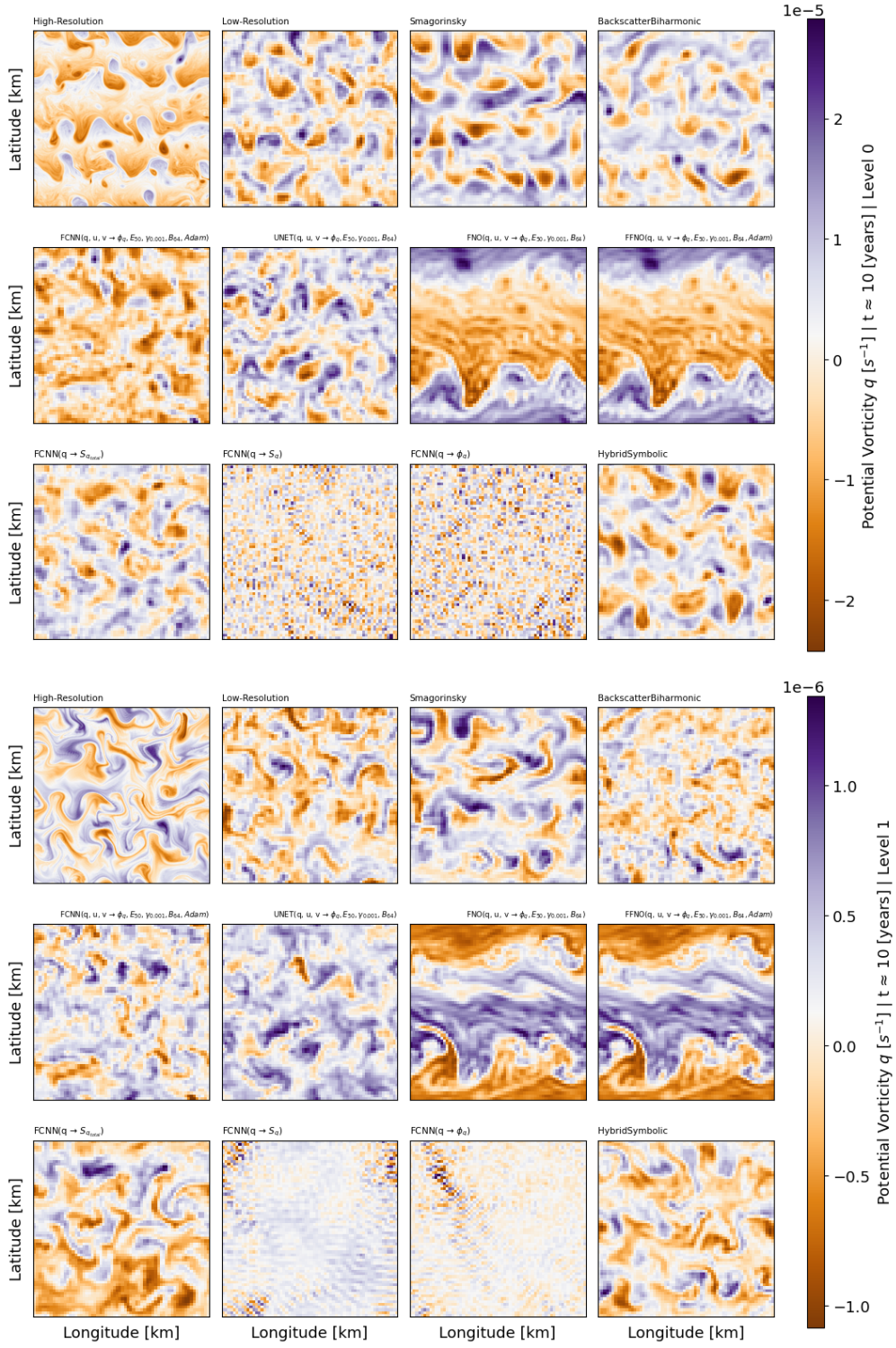


Figure 24: | **Online - Phase 1 - Potential vorticity** | Visualization of potential vorticity q for both upper (first three rows) and lower (last three rows) layers across different simulation types, indicated at the top of each image. Each image represents the q value spanning the entire computational domain after 10 years of simulations. The objective is to emphasize and visualize simulations that lose their physical relevance, becoming mere pixel grids, and to illustrate the divergence from the high-resolution simulation. Furthermore, the evaluated parameterizations are detailed in Tab.2, they were trained using *UJ5000* and tested on **jets online**.

PHASE II

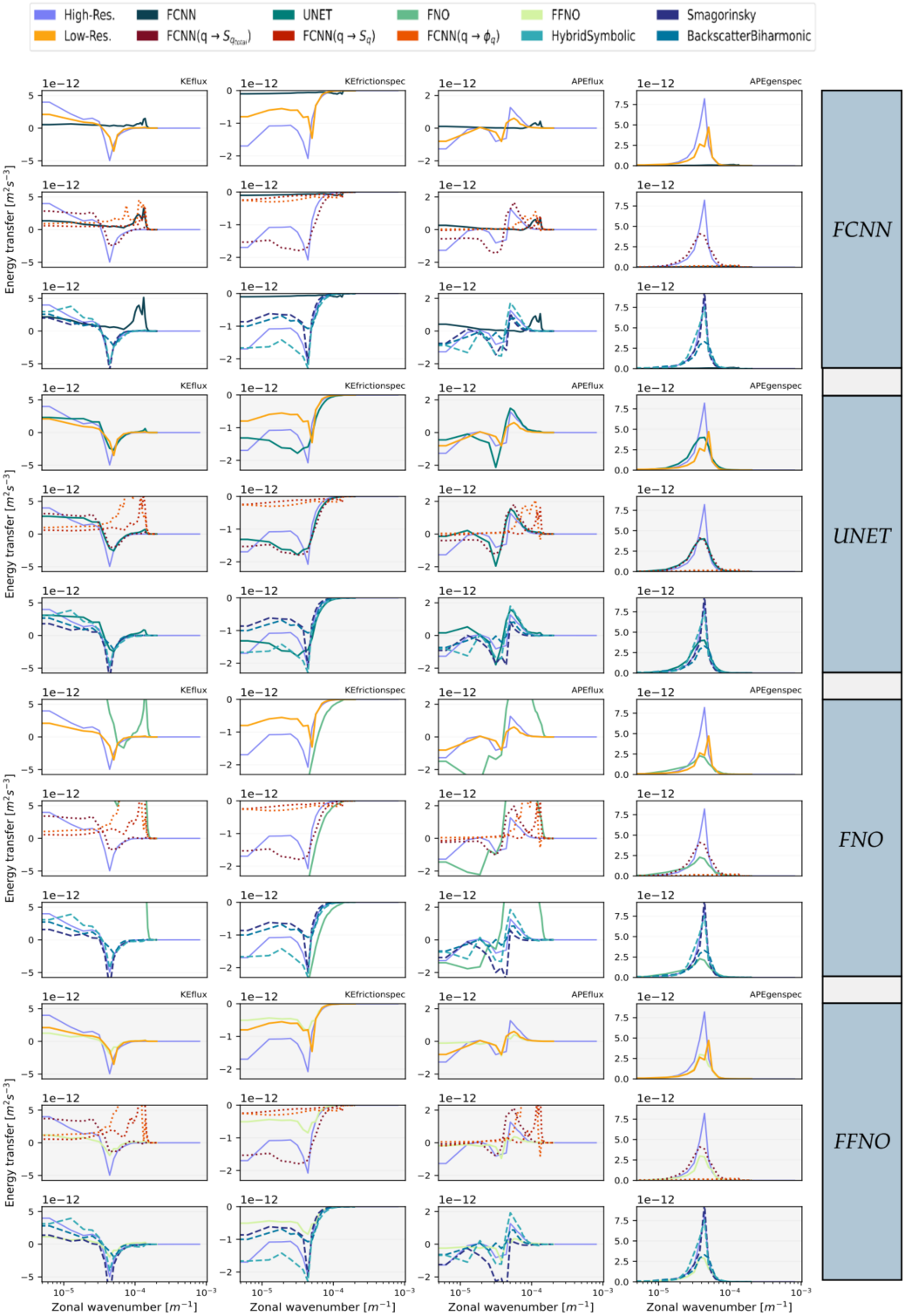


Figure 25: | **Online - Phase 2 - Energy budget** | This table displays energy spectra for **KEflux**, **KEfrictionspec**, **APEflux**, and **APEgenspec** using parameterizations grouped in Tab.3, these were trained on *MJ5000* and tested on **eddies online**. Each parameterization spectrum is compared against high-resolution and various low-resolution simulations, including neural networks from Ross et al., 2023 and analytical parameterizations from Smagorinsky, 1963; Jansen and Held, 2014; Zanna and Bolton, 2020.

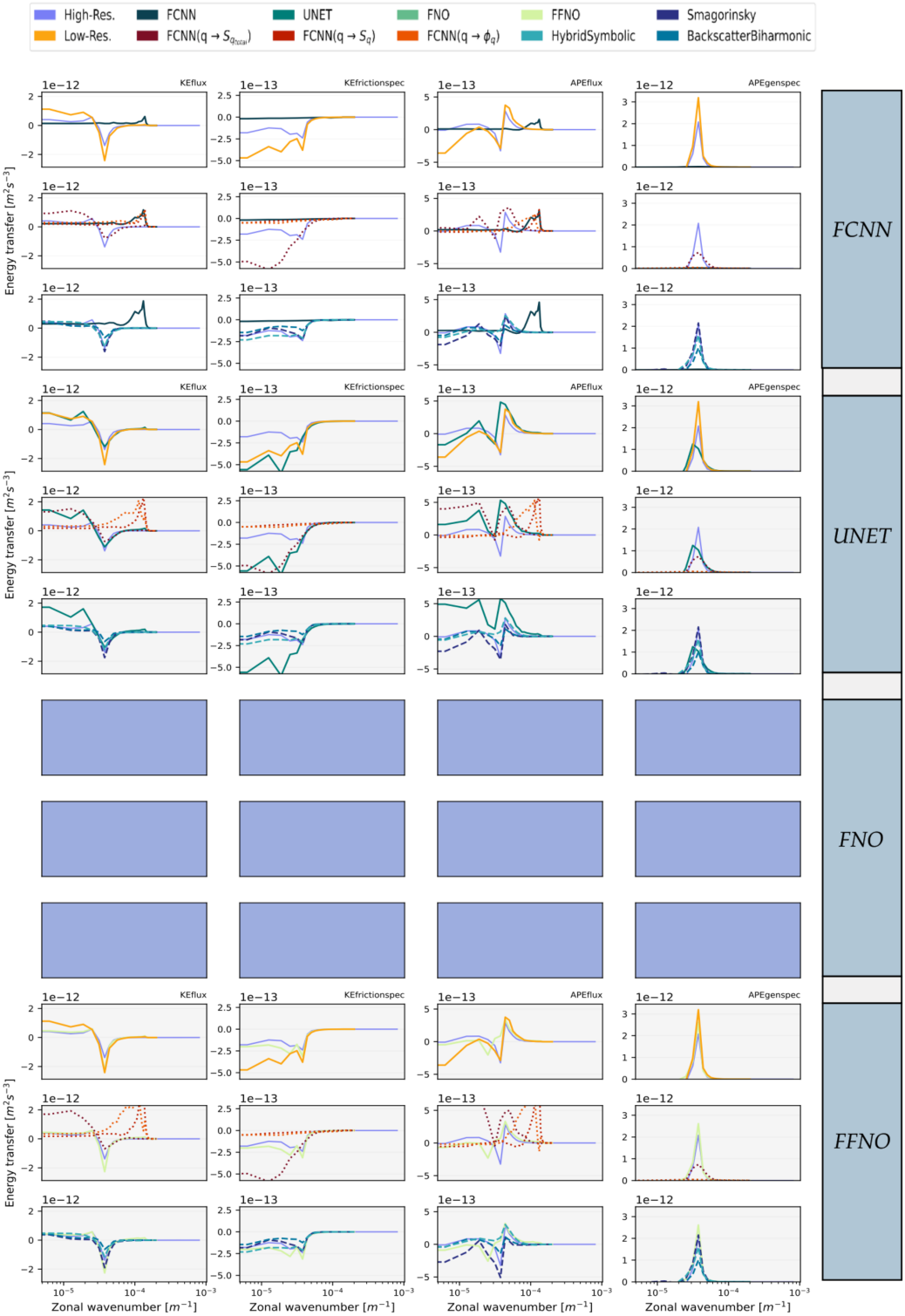


Figure 26: | **Online - Phase 2 - Energy budget** | This table displays energy spectra for **KEflux**, **KEfrictionspec**, **APEflux**, and **APEgenspec** using parameterizations grouped in Tab.3, these were trained on *MJ5000* and tested on **jets online**. Each parameterization spectrum is compared against high-resolution and various low-resolution simulations, including neural networks from Ross et al., 2023 and analytical parameterizations from Smagorinsky, 1963; Jansen and Held, 2014; Zanna and Bolton, 2020.

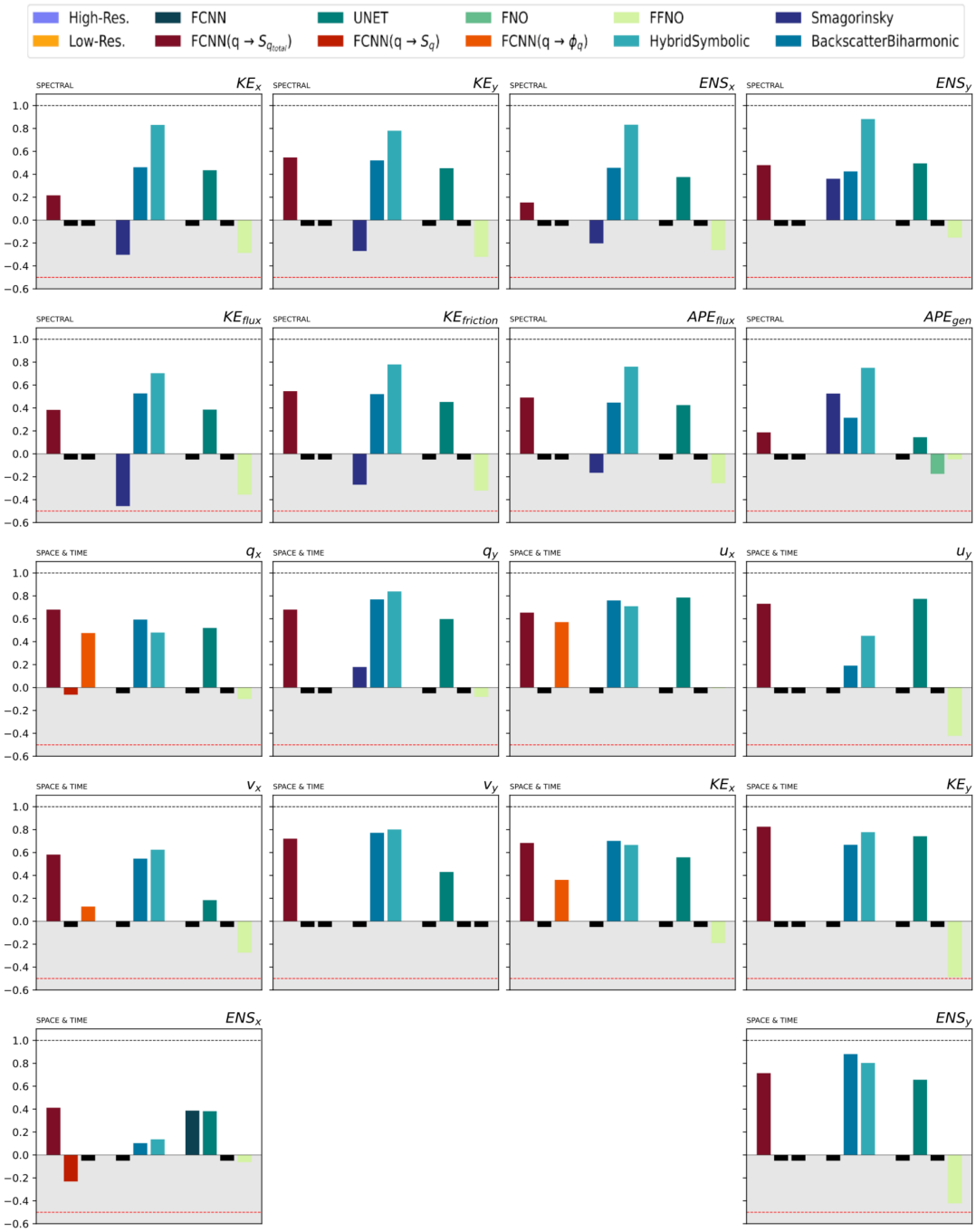


Figure 27: | **Online - Phase 2 - Similarities** | This table provides a summary of the Earth mover's distance, reformulated as a similarity metric for various flow quantities represented in either spectral or spatiotemporal domains. A value approaching 1 indicates strong agreement between the distribution obtained from high-resolution simulations and the current observations. Negative values are considered unfavorable, and values lower than -0.5 are disregarded. The tested parameterizations are grouped in Tab.3, they are trained on *MJ5000* and tested on **eddies online**. For comparison, the results of neural networks (Ross et al., 2023) and analytical parameterizations are also presented (Smagorinsky, 1963; Jansen and Held, 2014; Zanna and Bolton, 2020).

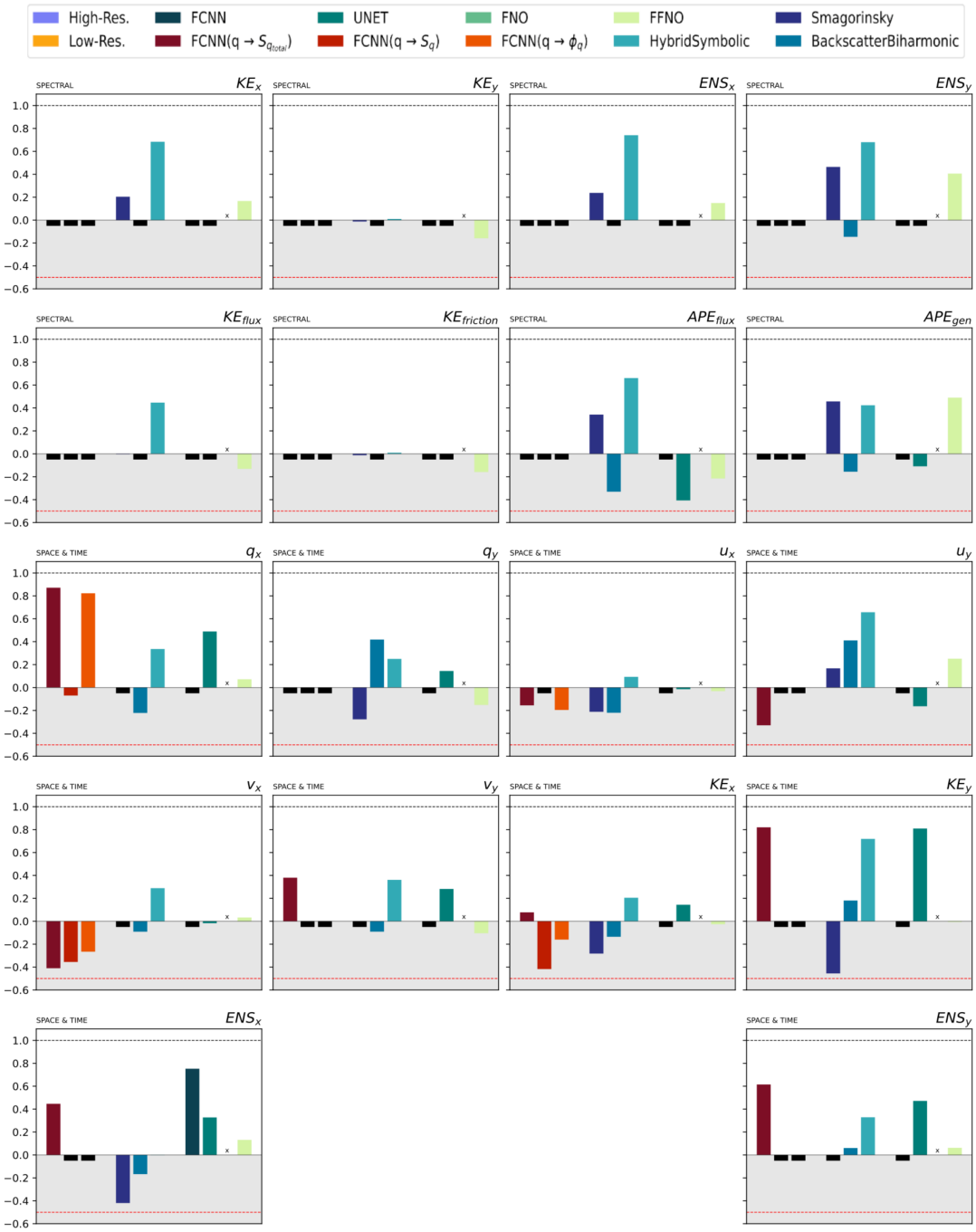


Figure 28: | **Online - Phase 2 - Similarities** | This table provides a summary of the Earth mover's distance, reformulated as a similarity metric for various flow quantities represented in either spectral or spatiotemporal domains. A value approaching 1 indicates strong agreement between the distribution obtained from high-resolution simulations and the current observations. Negative values are considered unfavorable, and values lower than -0.5 are disregarded. The tested parameterizations are grouped in Tab.3, they are trained on *MJ5000* and tested on **jets online**. For comparison, the results of neural networks (Ross et al., 2023) and analytical parameterizations are also presented (Smagorinsky, 1963; Jansen and Held, 2014; Zanna and Bolton, 2020).

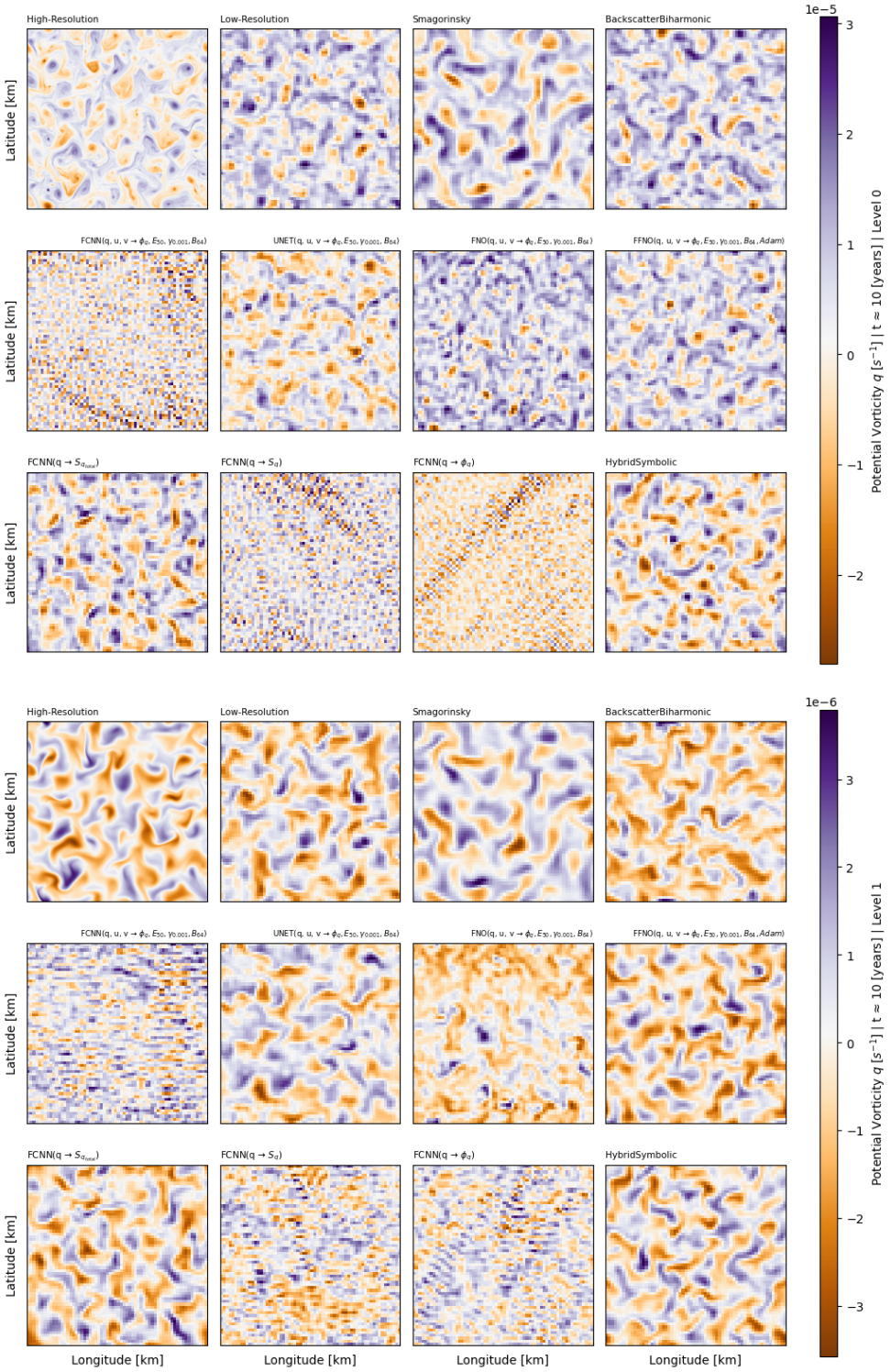


Figure 29: | **Online - Phase 2 - Potential vorticity** | Visualization of potential vorticity q for both upper (first three rows) and lower (last three rows) layers across different simulation types, indicated at the top of each image. Each image represents the q value spanning the entire computational domain after 10 years of simulations. The objective is to emphasize and visualize simulations that lose their physical relevance, becoming mere pixel grids, and to illustrate the divergence from the high-resolution simulation. Furthermore, the evaluated parameterizations are detailed in Tab.3, they were trained using *MJ5000* and tested on **eddies online**.

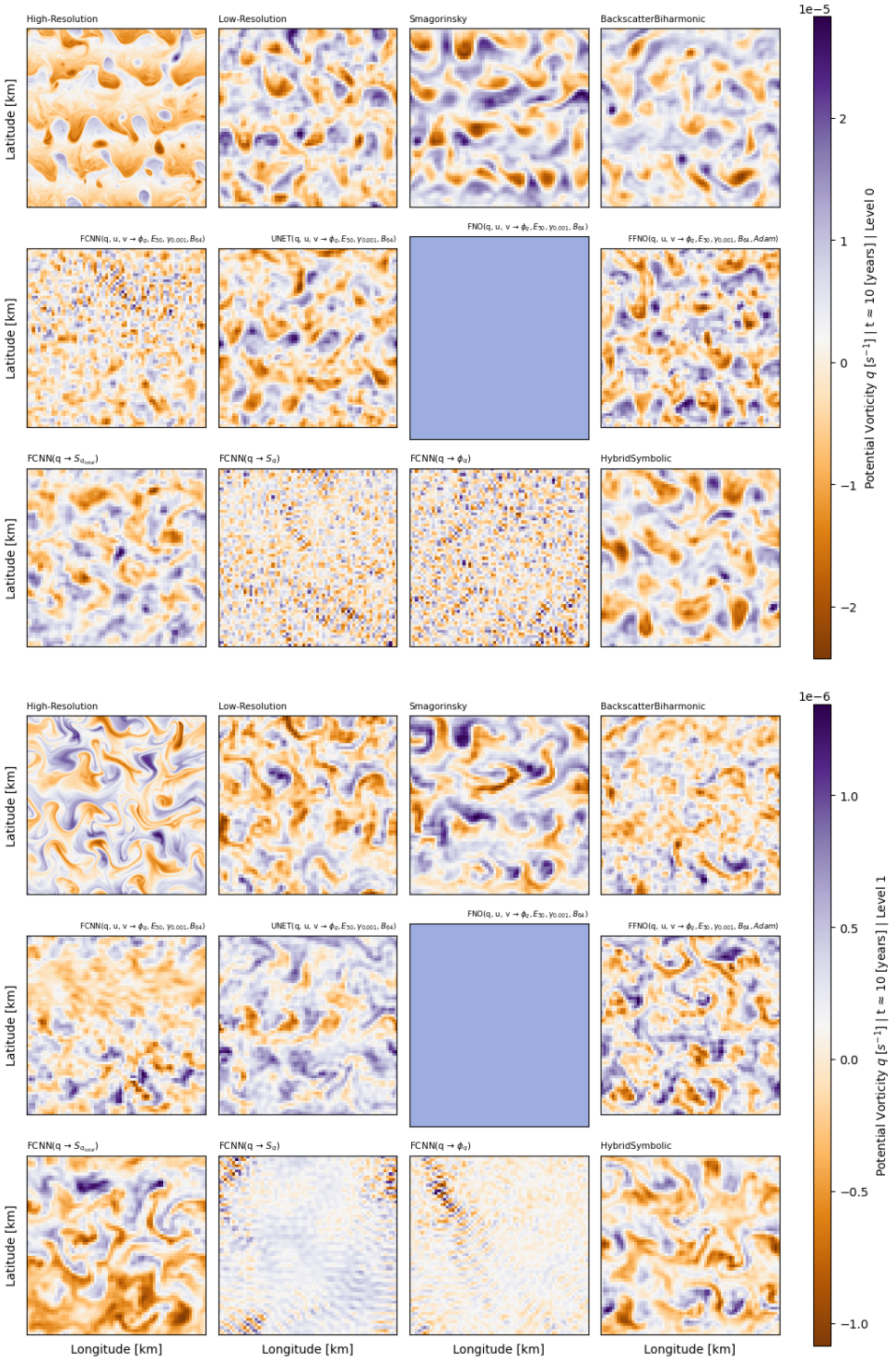


Figure 30: | **Online - Phase 2 - Potential vorticity** | Visualization of potential vorticity q for both upper (first three rows) and lower (last three rows) layers across different simulation types, indicated at the top of each image. Each image represents the q value spanning the entire computational domain after 10 years of simulations. The objective is to emphasize and visualize simulations that lose their physical relevance, becoming mere pixel grids, and to illustrate the divergence from the high-resolution simulation. Furthermore, the evaluated parameterizations are detailed in Tab.3, they were trained using *MJ5000* and tested on **jets online**.

PHASE III

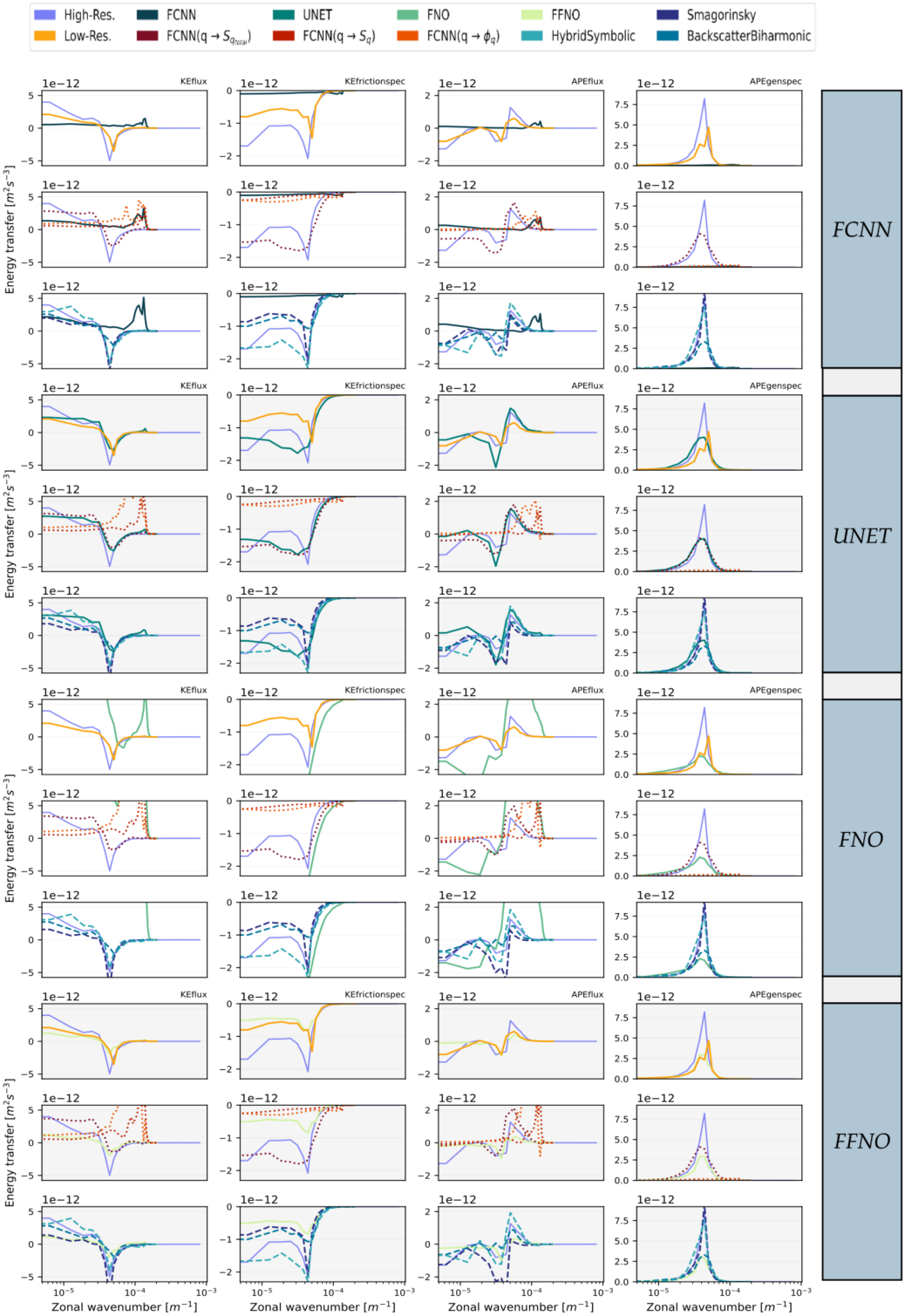


Figure 31: | **Online - Phase 3 - Energy budget** | This table displays energy spectra for **KEflux**, **KEfrictionspec**, **APEflux**, and **APEgenspec** using parameterizations grouped in Tab.2, these were trained on *MJ20000* and tested on **eddies online**. Each parameterization spectrum is compared against high-resolution and various low-resolution simulations, including neural networks from Ross et al., 2023 and analytical parameterizations from Smagorinsky, 1963; Jansen and Held, 2014; Zanna and Bolton, 2020.

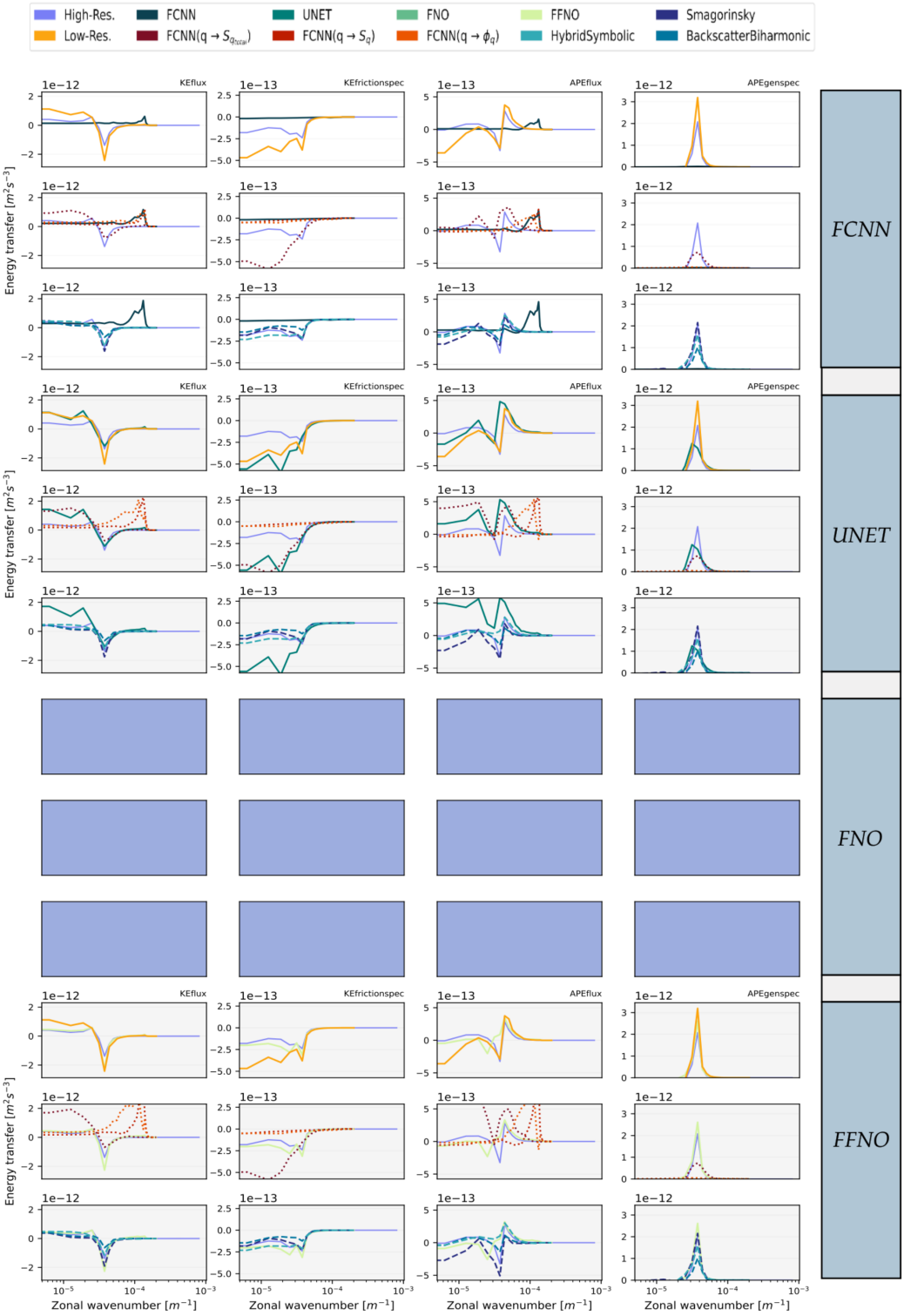


Figure 32: | **Online - Phase 3 - Energy budget** | This table displays energy spectra for **KEflux**, **KEfrictionspec**, **APEflux**, and **APEgenspec** using parameterizations grouped in Tab.2, these were trained on *MJ20000* and tested on **jets online**. Each parameterization spectrum is compared against high-resolution and various low-resolution simulations, including neural networks from Ross et al., 2023 and analytical parameterizations from Smagorinsky, 1963; Jansen and Held, 2014; Zanna and Bolton, 2020.

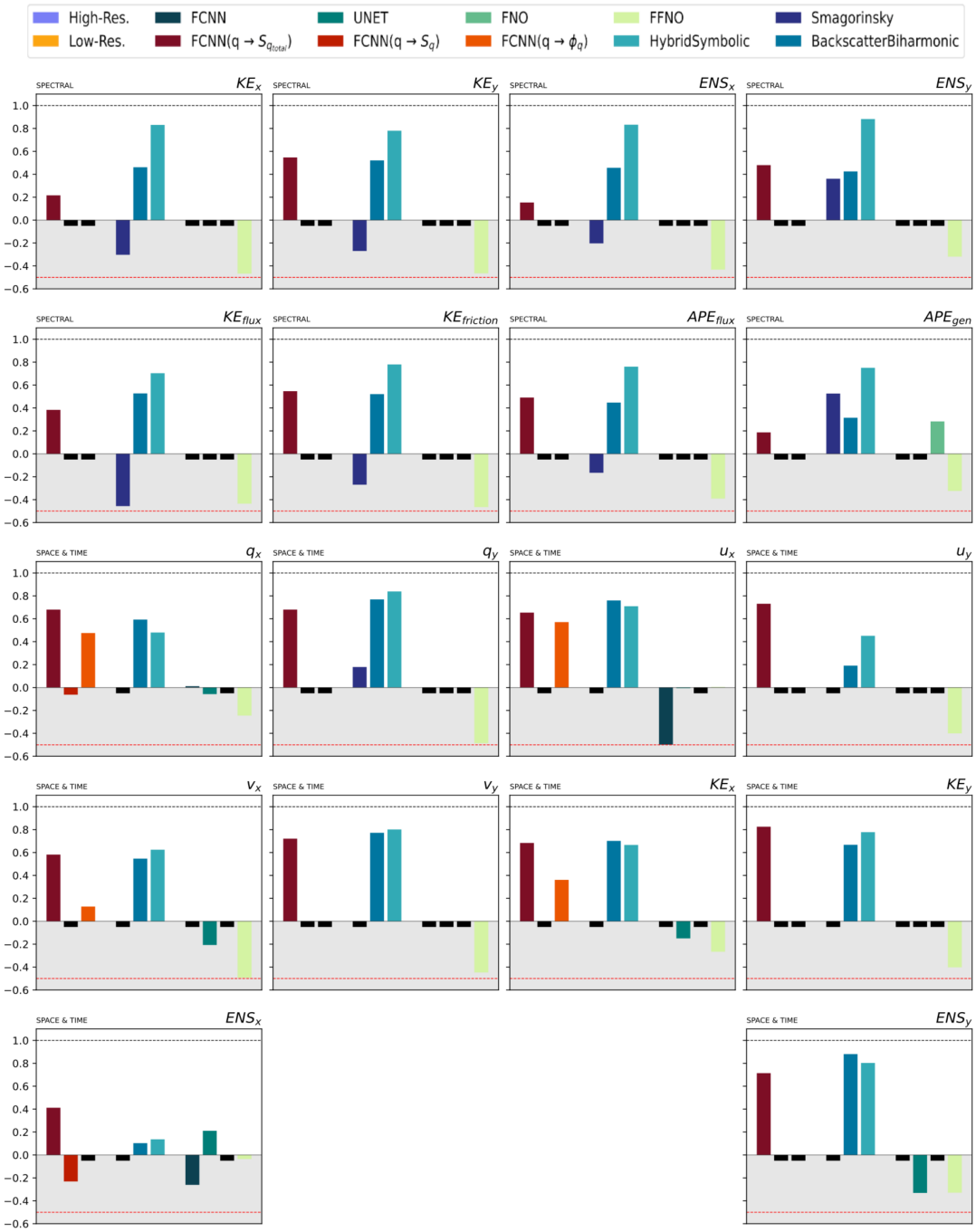


Figure 33: | **Online - Phase 3 - Similarities** | This table provides a summary of the Earth mover's distance, reformulated as a similarity metric for various flow quantities represented in either spectral or spatiotemporal domains. A value approaching 1 indicates strong agreement between the distribution obtained from high-resolution simulations and the current observations. Negative values are considered unfavorable, and values lower than -0.5 are disregarded. The tested parameterizations are grouped in Tab.2, they are trained on *MJ20000* and tested on **eddies online**. For comparison, the results of neural networks (Ross et al., 2023) and analytical parameterizations are also presented (Smagorinsky, 1963; Jansen and Held, 2014; Zanna and Bolton, 2020).

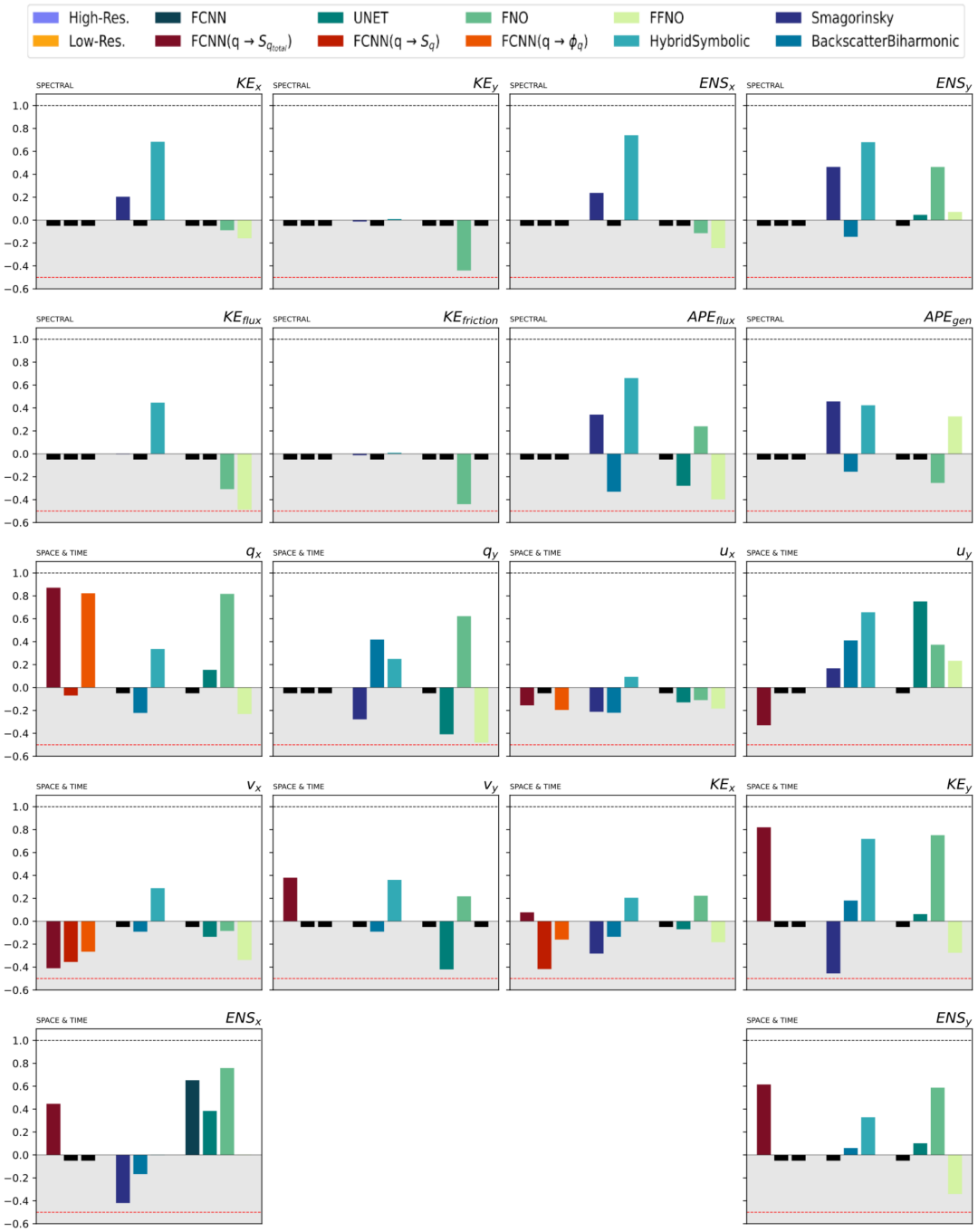


Figure 34: | **Online - Phase 3 - Similarities** | This table provides a summary of the Earth mover's distance, reformulated as a similarity metric for various flow quantities represented in either spectral or spatiotemporal domains. A value approaching 1 indicates strong agreement between the distribution obtained from high-resolution simulations and the current observations. Negative values are considered unfavorable, and values lower than -0.5 are disregarded. The tested parameterizations are grouped in Tab.2, they are trained on *MJ20000* and tested on **jets online**. For comparison, the results of neural networks (Ross et al., 2023) and analytical parameterizations are also presented (Smagorinsky, 1963; Jansen and Held, 2014; Zanna and Bolton, 2020).

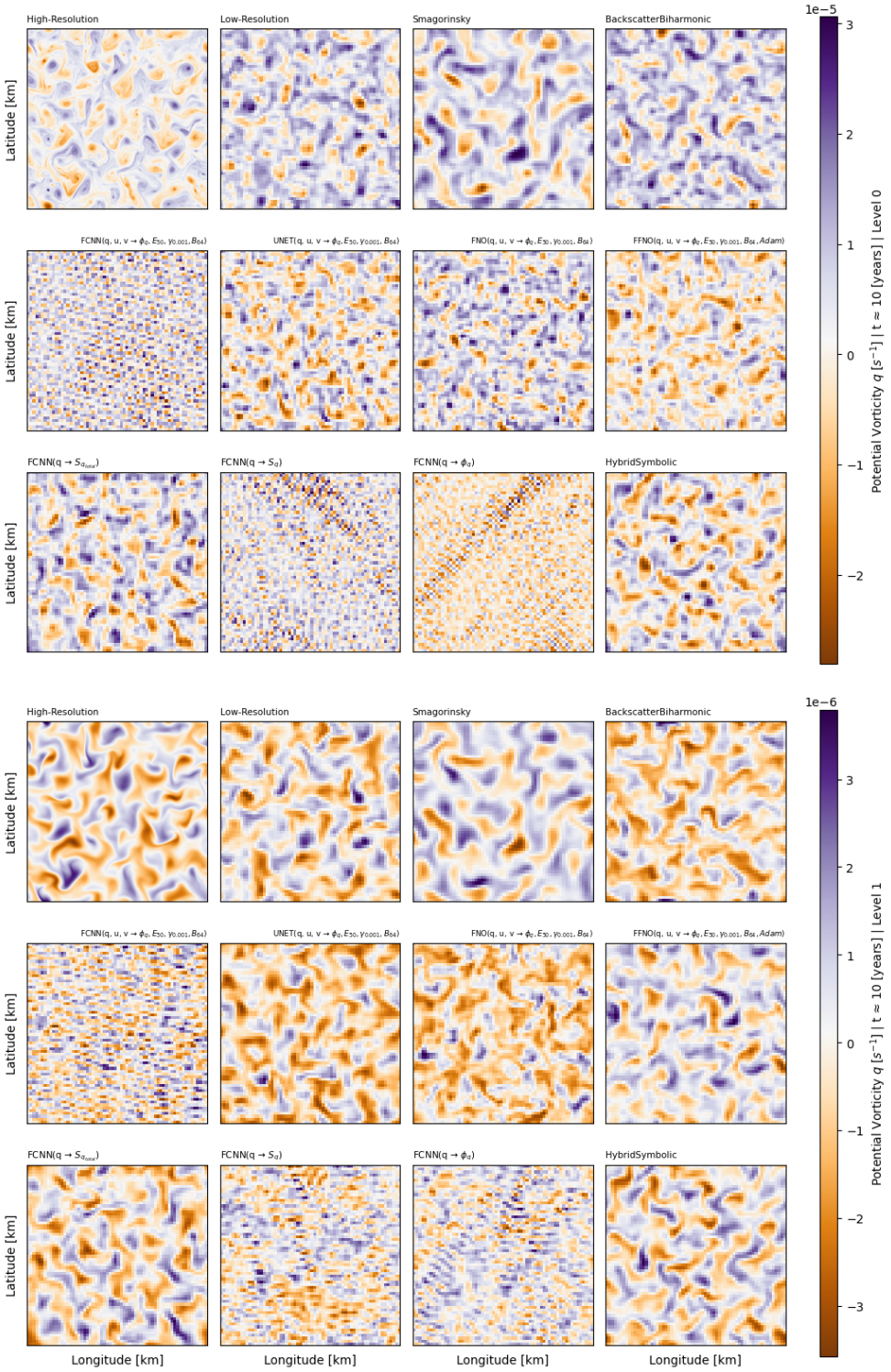


Figure 35: | **Online - Phase 3 - Potential vorticity** | Visualization of potential vorticity q for both upper (first three rows) and lower (last three rows) layers across different simulation types, indicated at the top of each image. Each image represents the q value spanning the entire computational domain after 10 years of simulations. The objective is to emphasize and visualize simulations that lose their physical relevance, becoming mere pixel grids, and to illustrate the divergence from the high-resolution simulation. Furthermore, the evaluated parameterizations are detailed in Tab.2, they were trained using *MJ20000* and tested on **eddies online**.

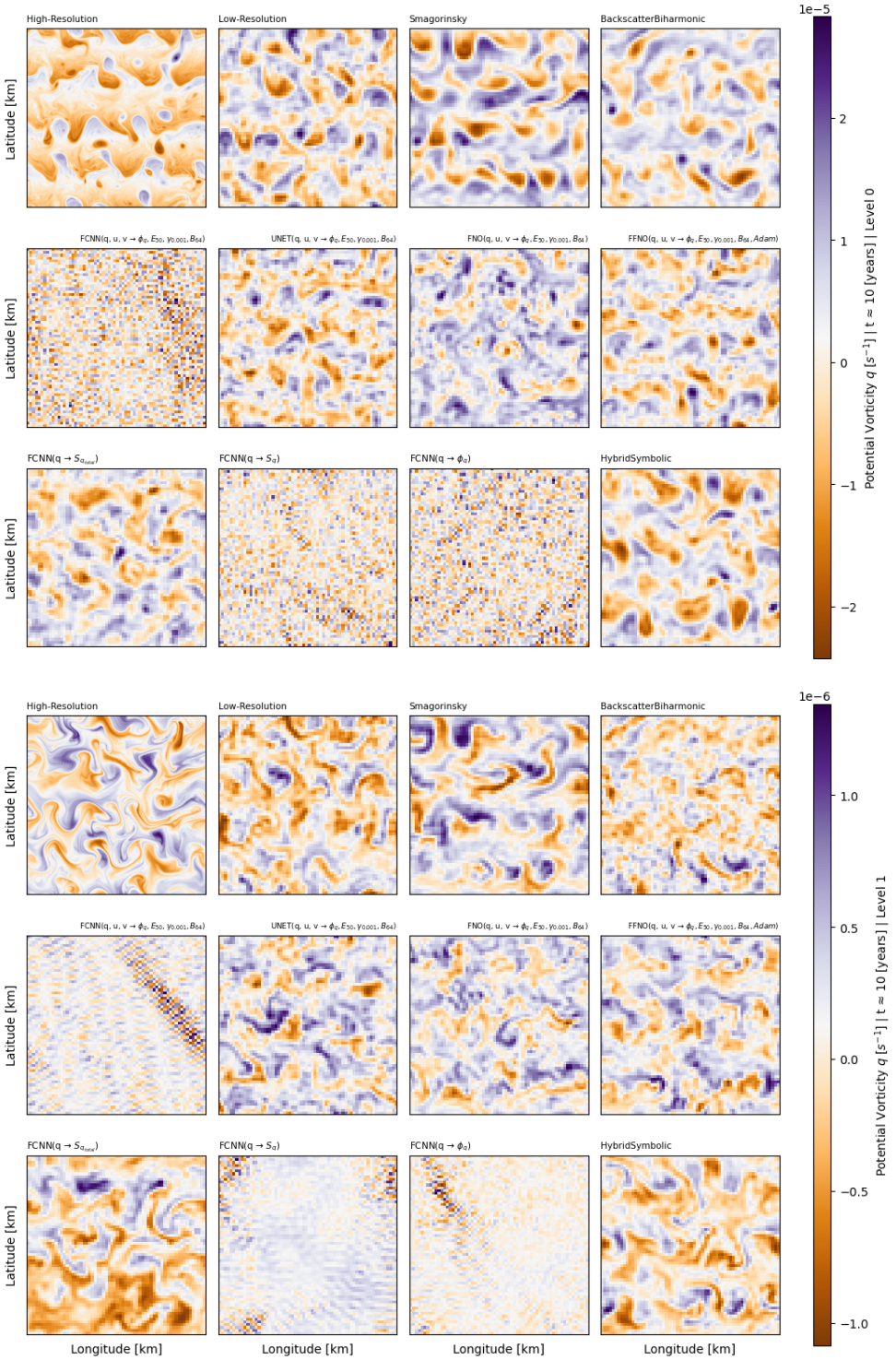


Figure 36: | **Online - Phase 3 - Potential vorticity** | Visualization of potential vorticity q for both upper (first three rows) and lower (last three rows) layers across different simulation types, indicated at the top of each image. Each image represents the q value spanning the entire computational domain after 10 years of simulations. The objective is to emphasize and visualize simulations that lose their physical relevance, becoming mere pixel grids, and to illustrate the divergence from the high-resolution simulation. Furthermore, the evaluated parameterizations are detailed in Tab.2, they were trained using *MJ20000* and tested on **jets online**.

PHASE IV

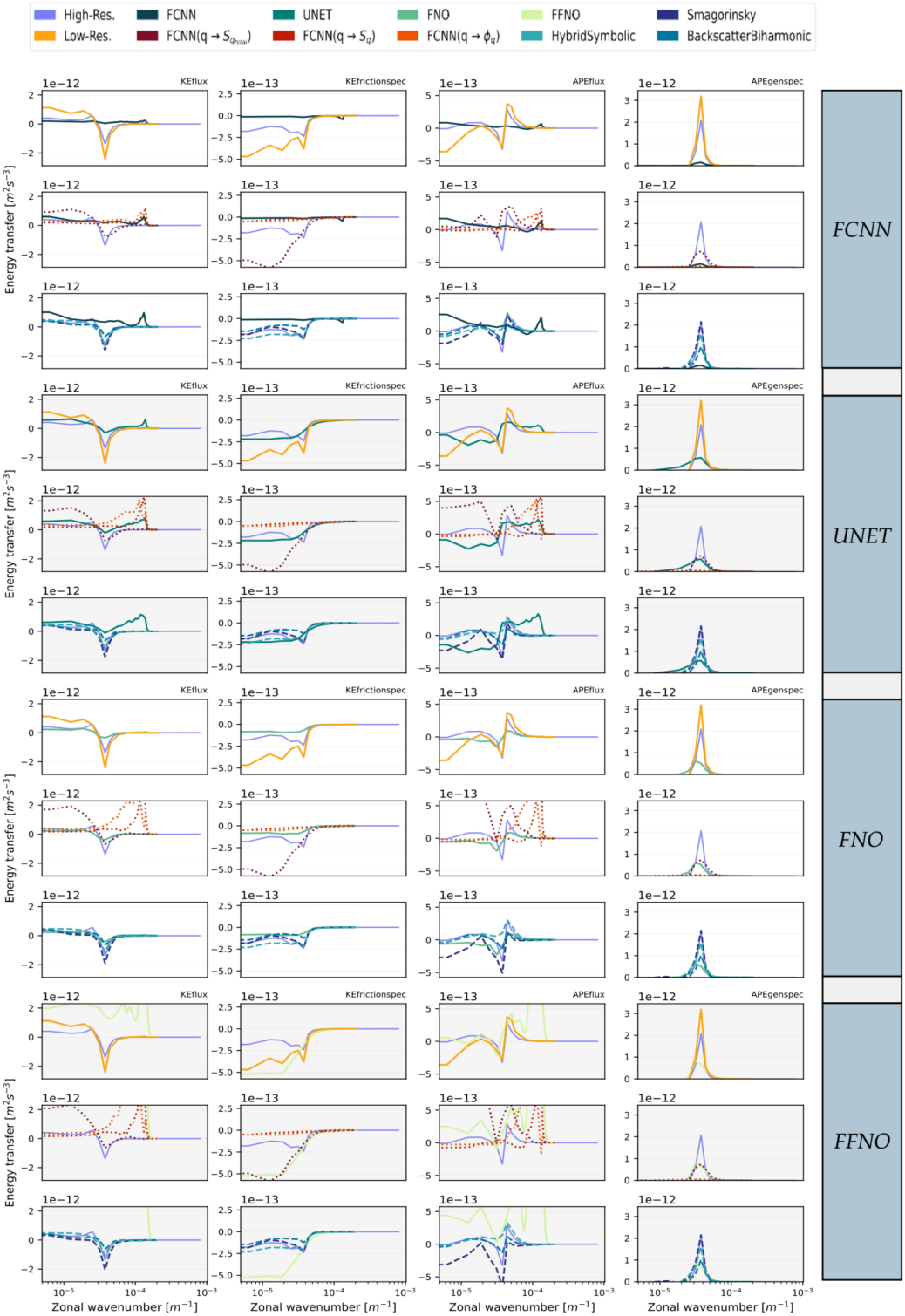


Figure 37: | **Online - Phase 4 - Energy budget** | This table displays energy spectra for **KEflux**, **KEfrictionspec**, **APEflux**, and **APEgenspec** using parameterizations grouped in Tab.5, these were trained on *F40000* and tested on **jets online**. Each parameterization spectrum is compared against high-resolution and various low-resolution simulations, including neural networks from Ross et al., 2023 and analytical parameterizations from Smagorinsky, 1963; Jansen and Held, 2014; Zanna and Bolton, 2020.

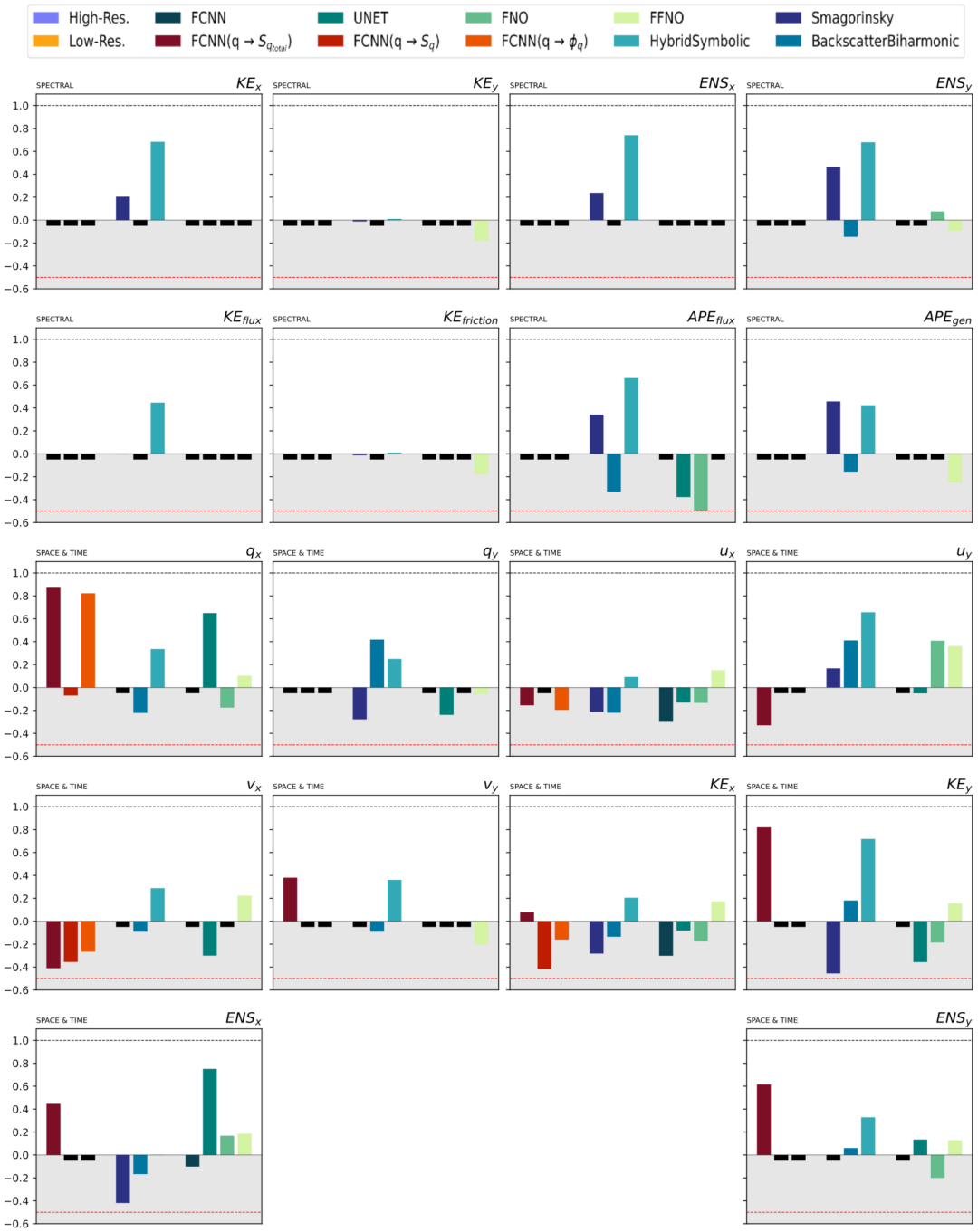


Figure 38: | **Online - Phase 4 - Similarities** | This table provides a summary of the Earth mover's distance, reformulated as a similarity metric for various flow quantities represented in either spectral or spatiotemporal domains. A value approaching 1 indicates strong agreement between the distribution obtained from high-resolution simulations and the current observations. Negative values are considered unfavorable, and values lower than -0.5 are disregarded. The tested parameterizations are grouped in Tab.5, they are trained on *F40000* and tested on **jets online**. For comparison, the results of neural networks (Ross et al., 2023) and analytical parameterizations are also presented (Smagorinsky, 1963; Jansen and Held, 2014; Zanna and Bolton, 2020).

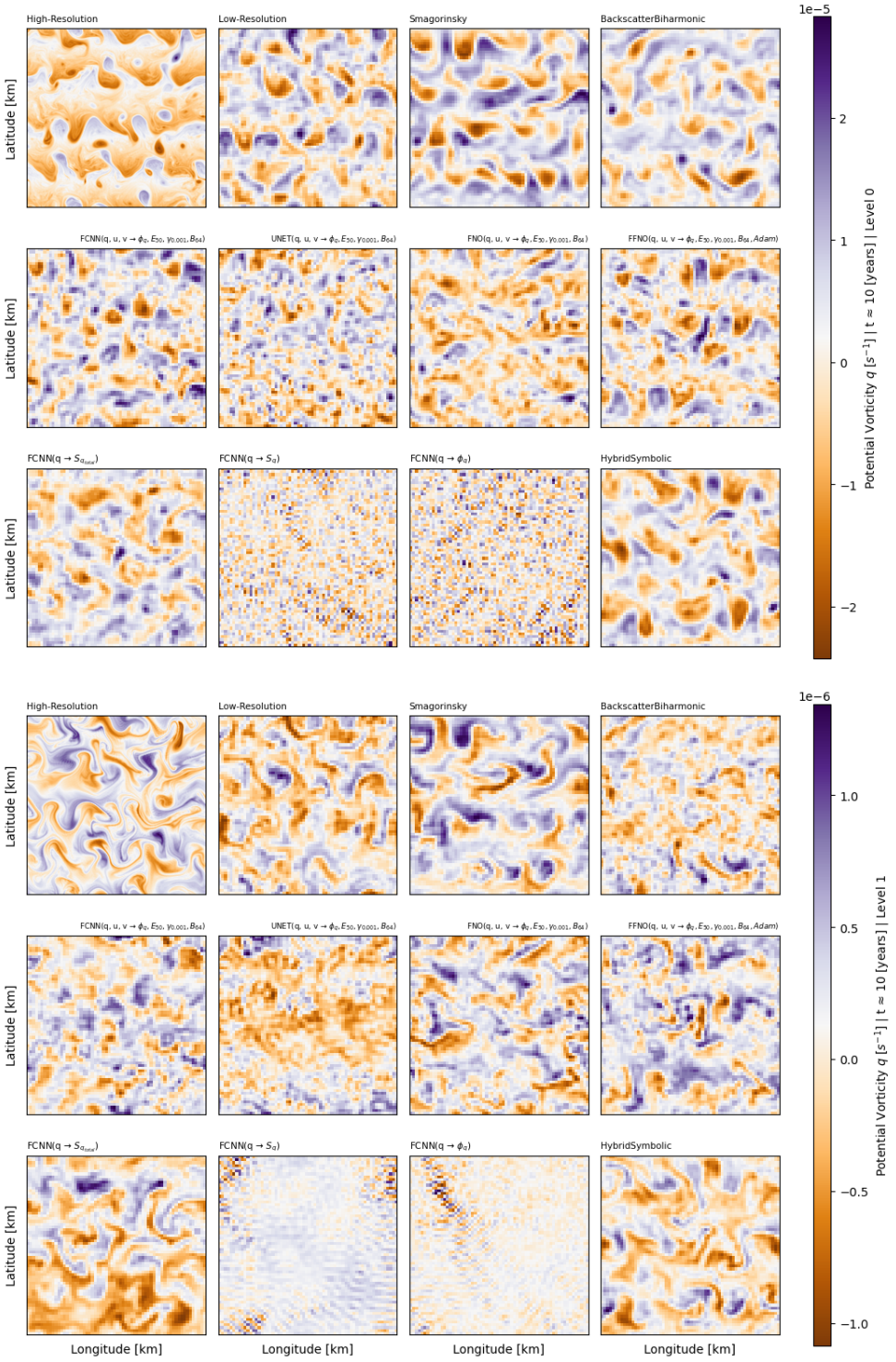


Figure 39: | **Online - Phase 4 - Potential vorticity** | Visualization of potential vorticity q for both upper (first three rows) and lower (last three rows) layers across different simulation types, indicated at the top of each image. Each image represents the q value spanning the entire computational domain after 10 years of simulations. The objective is to emphasize and visualize simulations that lose their physical relevance, becoming mere pixel grids, and to illustrate the divergence from the high-resolution simulation. Furthermore, the evaluated parameterizations are detailed in Tab.5, they were trained using *F40000* and tested on **jets online**.

PHASE V

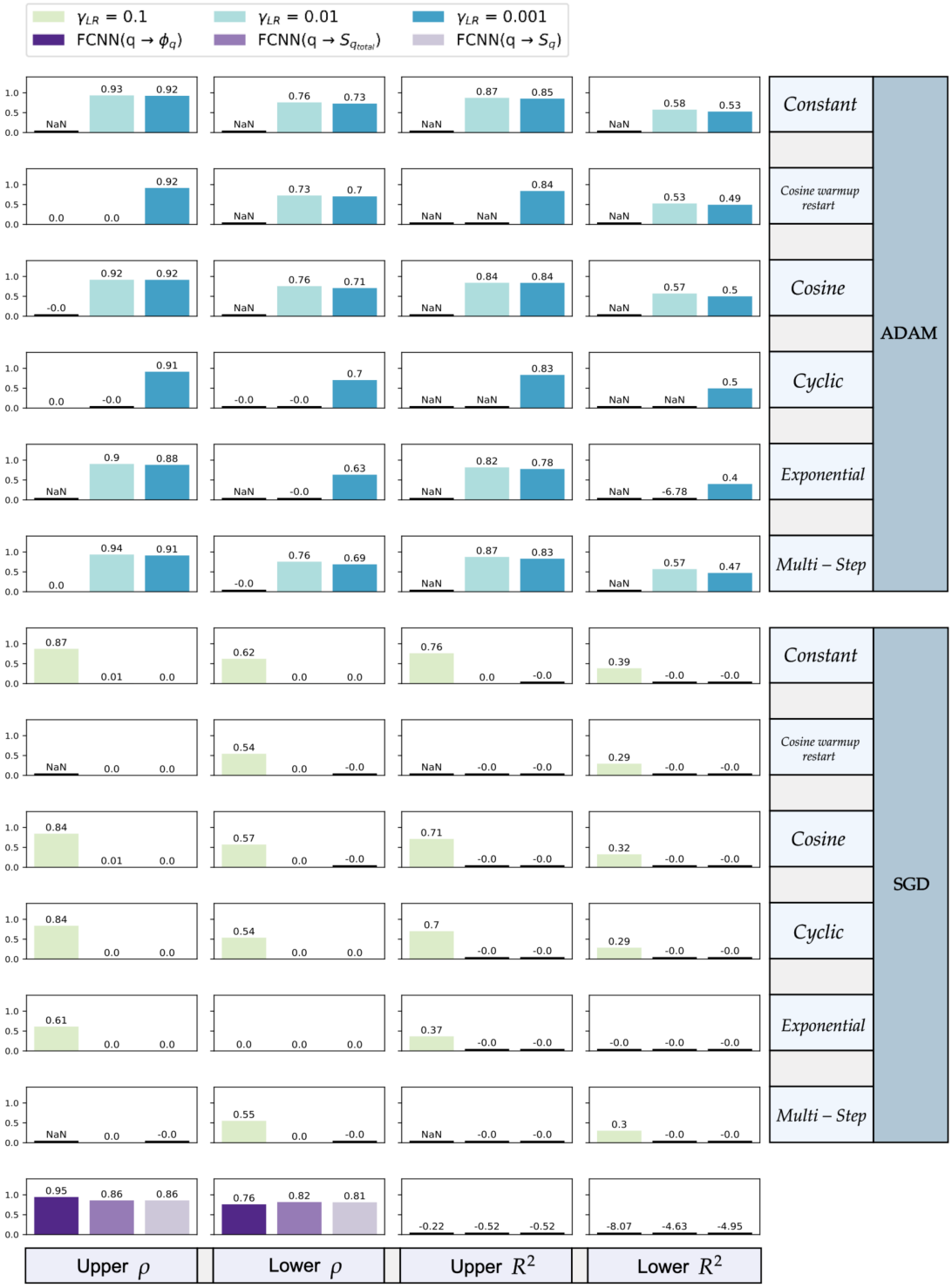


Figure 40: | **Offline - Phase 5** | This table summarize offline results, including correlations (columns 1 and 2) and mean-squared errors (columns 3 and 4), for parameterizations trained on **full dataset 5000**, evaluated on dataset **jets offline** and predicting subgrid flux $\mathbf{\Phi}_q$ (see Eq. 23). On the right, details regarding the optimizer and scheduler employed for training are provided, along with the corresponding learning rates displayed in the legend. The bottom row presents results obtained using the three FCNN parameterizations introduced in Ross et al., 2023.

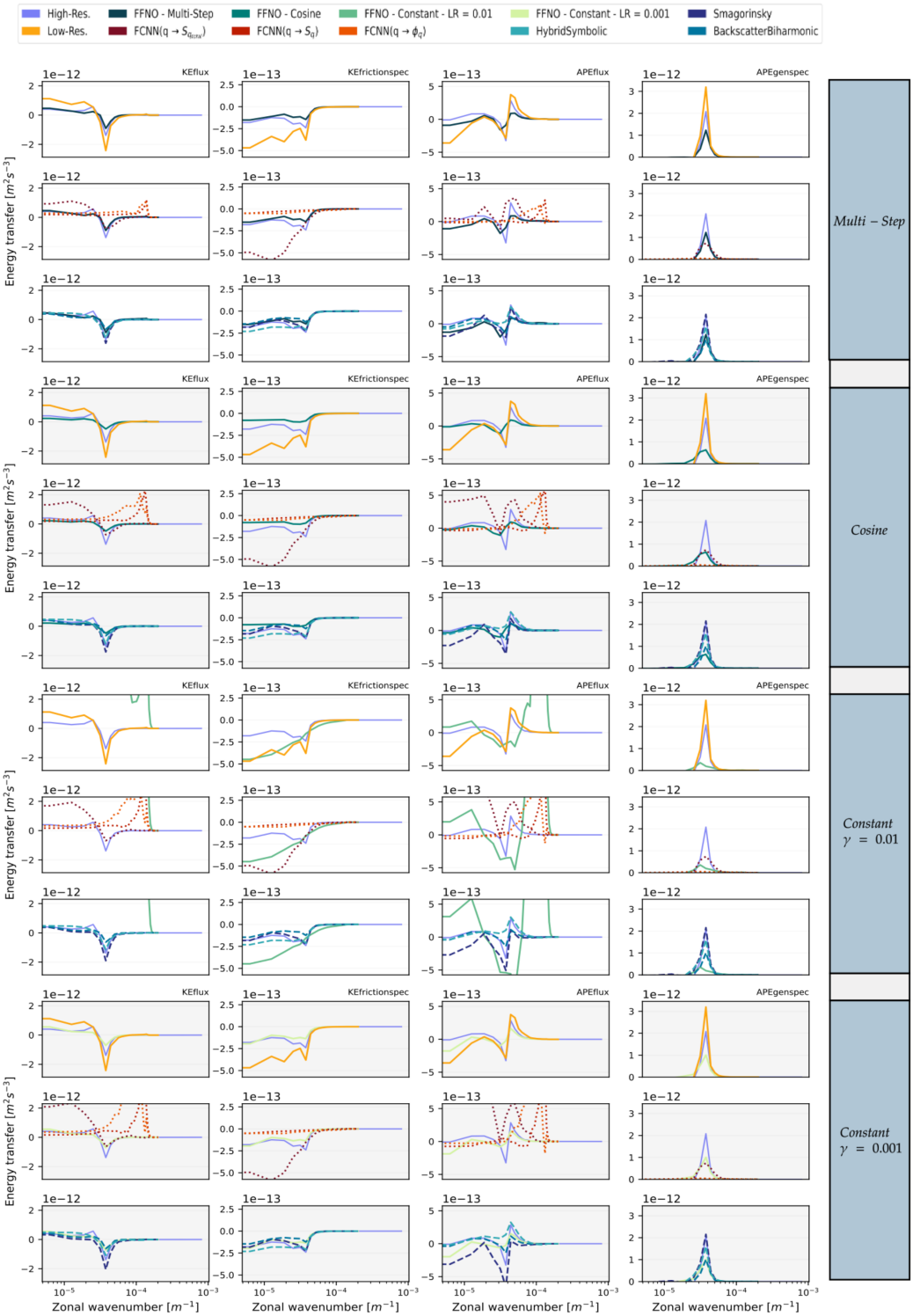


Figure 41: | **Online - Phase 5 - Energy budget** | This table displays energy spectra for **KEflux**, **KEfrictionspec**, **APEflux**, and **APEgenspec** using parameterizations of Tab.6, these were trained on *F5000* and tested on **jets online**. Each parameterization spectrum is compared against high-resolution and various low-resolution simulations, including neural networks from Ross et al., 2023 and analytical parameterizations from Smagorinsky, 1963; Jansen and Held, 2014; Zanna and Bolton, 2020.

Figure 42: | **Online - Phase 5 - Similarities** | This table provides a summary of the Earth mover's distance, reformulated as a similarity metric for various flow quantities represented in either spectral or spatiotemporal domains. A value approaching 1 indicates strong agreement between the distribution obtained from high-resolution simulations and the current observations. Negative values are considered unfavorable, and values lower than -0.5 are disregarded. The tested parameterizations comes from Tab.6, they are trained on *F5000* and tested on **jets online**. For comparison, the results of neural networks (Ross et al., 2023) and analytical parameterizations are also presented (Smagorinsky, 1963; Jansen and Held, 2014; Zanna and Bolton, 2020).

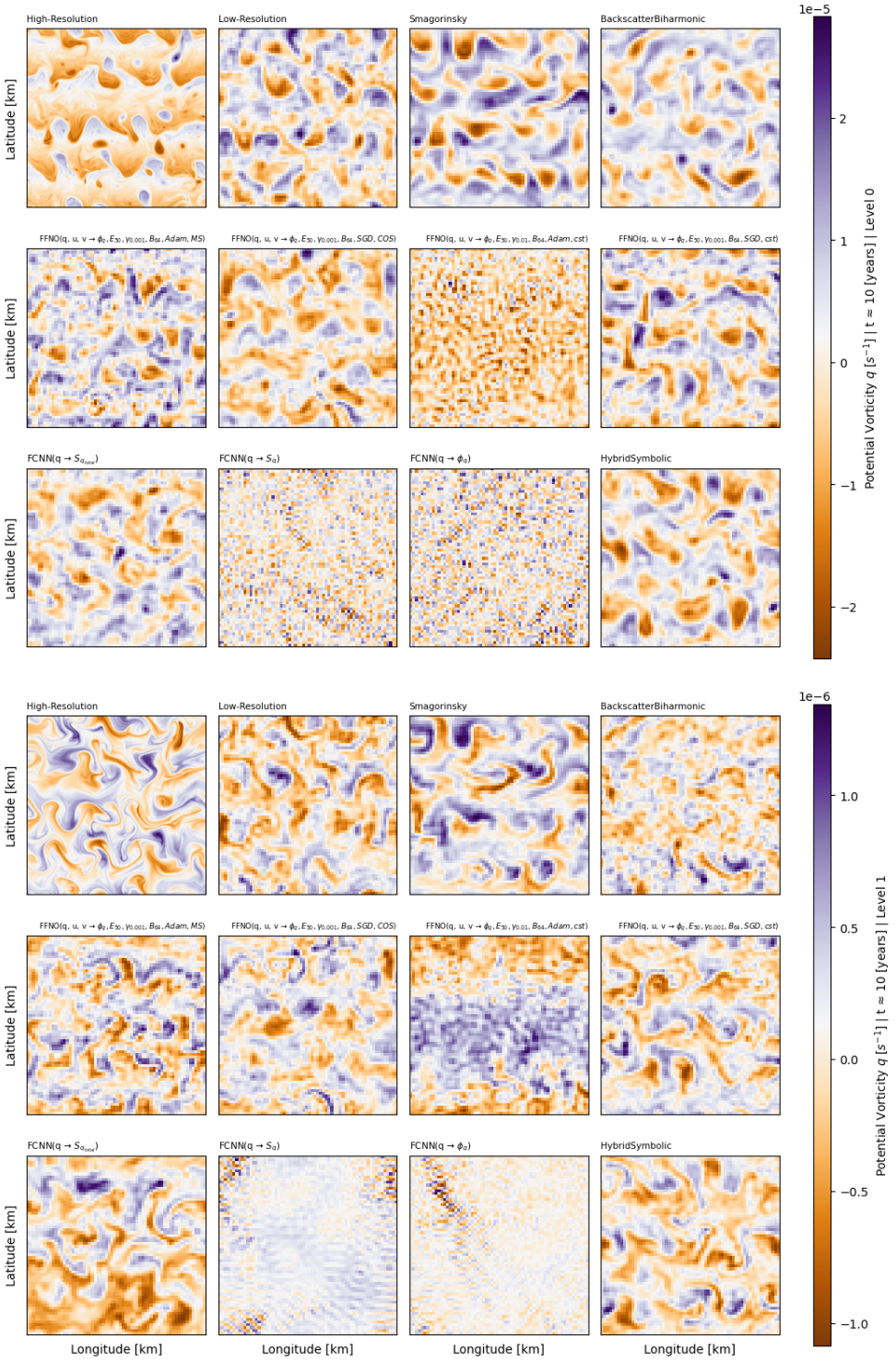


Figure 43: | **Online - Phase 5 - Potential vorticity** | Visualization of potential vorticity is presented for both upper (first three rows) and lower (last three rows) layers across different simulation types, indicated at the top of each image. Each image represents the q value spanning the entire computational domain after 10 years of simulations. The objective is to emphasize and visualize simulations that lose their physical relevance, becoming mere pixel grids, and to illustrate the divergence from the high-resolution simulation. Furthermore, the evaluated parameterizations are detailed in Tab.6, they were trained using *F5000* and assessed against **jets online**.

PHASE VI

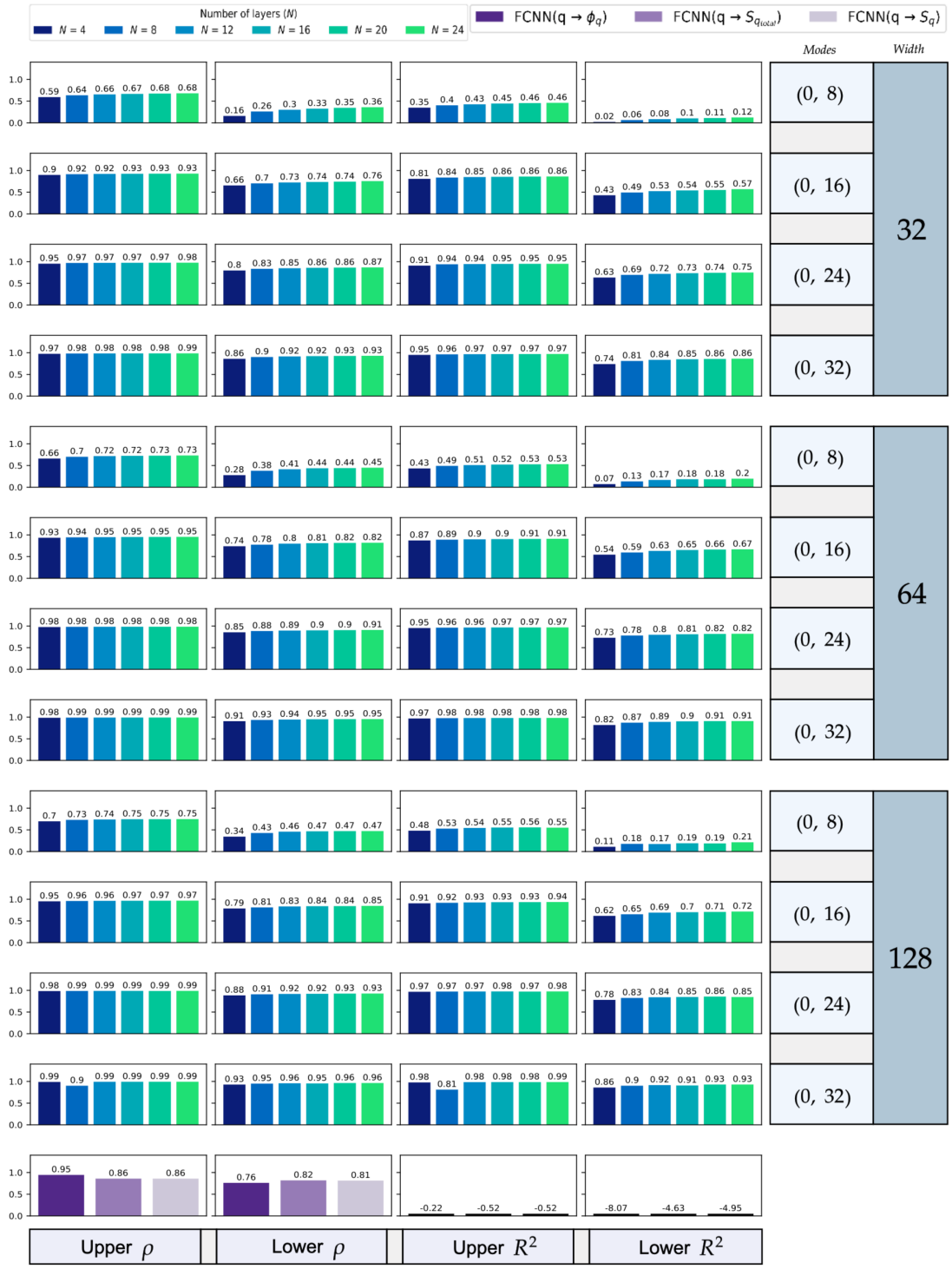


Figure 44: | **Offline - Phase 6 - Part 1** | This table summarize offline results, including correlations (columns 1 and 2) and mean-squared errors (columns 3 and 4), for parameterizations trained on *F5000*, evaluated on dataset **jets offline** and predicting subgrid flux $\mathbf{\Phi}_q$ (see Eq. 23). On the right-hand side, the width value, retained Fourier modes, and total number of layers for the FFNO are provided in the legend. The bottom row presents results obtained using the three FCNN parameterizations introduced in Ross et al., 2023.

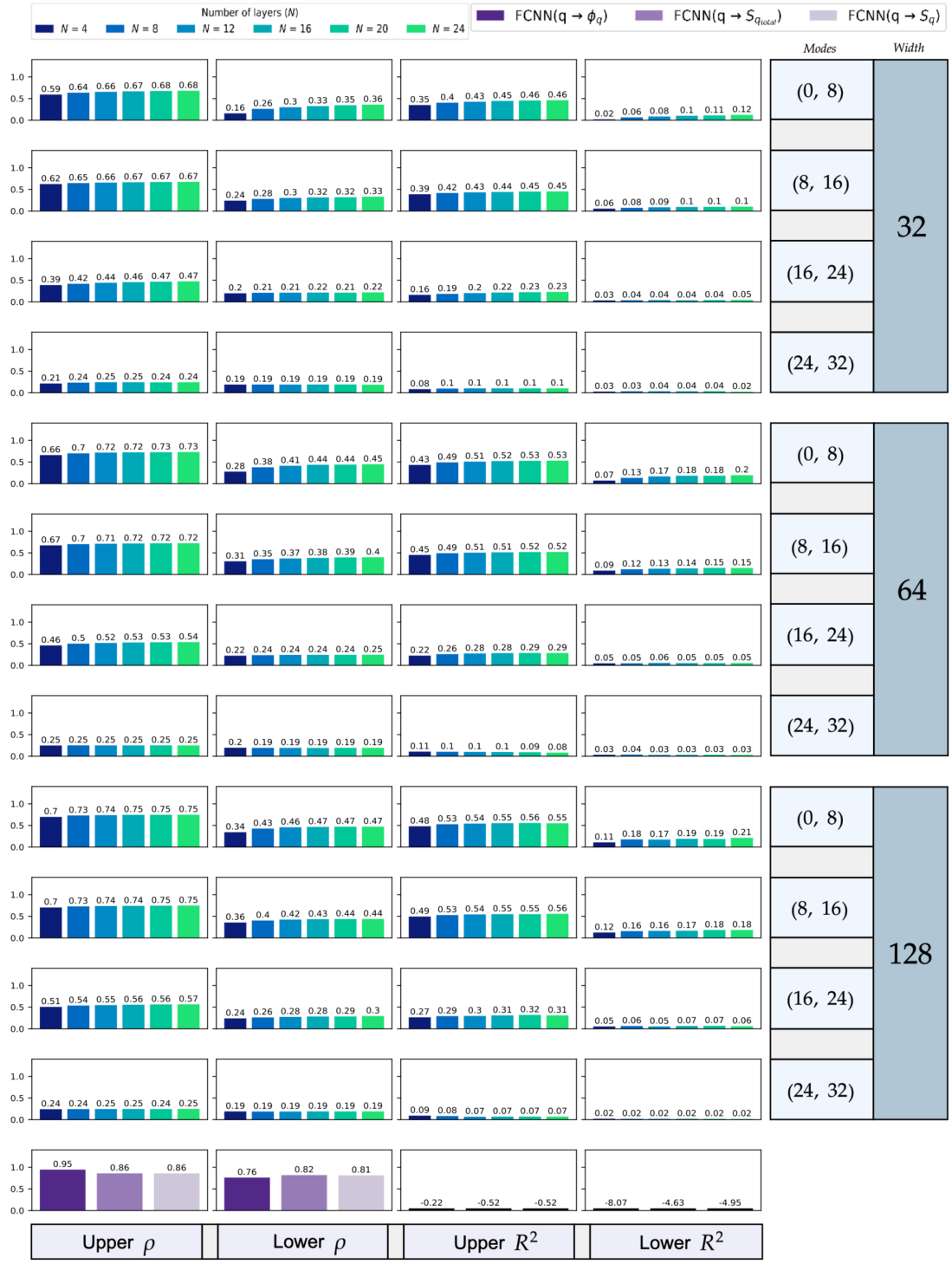


Figure 45: | **Offline - Phase 6 - Part 2** | This table summarize offline results, including correlations (columns 1 and 2) and mean-squared errors (columns 3 and 4), for parameterizations trained on *F5000*, evaluated on dataset **jets offline** and predicting subgrid flux $\mathbf{\Phi}_q$ (see Eq. 23). On the right-hand side, the width value, retained Fourier modes, and total number of layers for the FFNO are provided in the legend. The bottom row presents results obtained using the three FCNN parameterizations introduced in Ross et al., 2023.

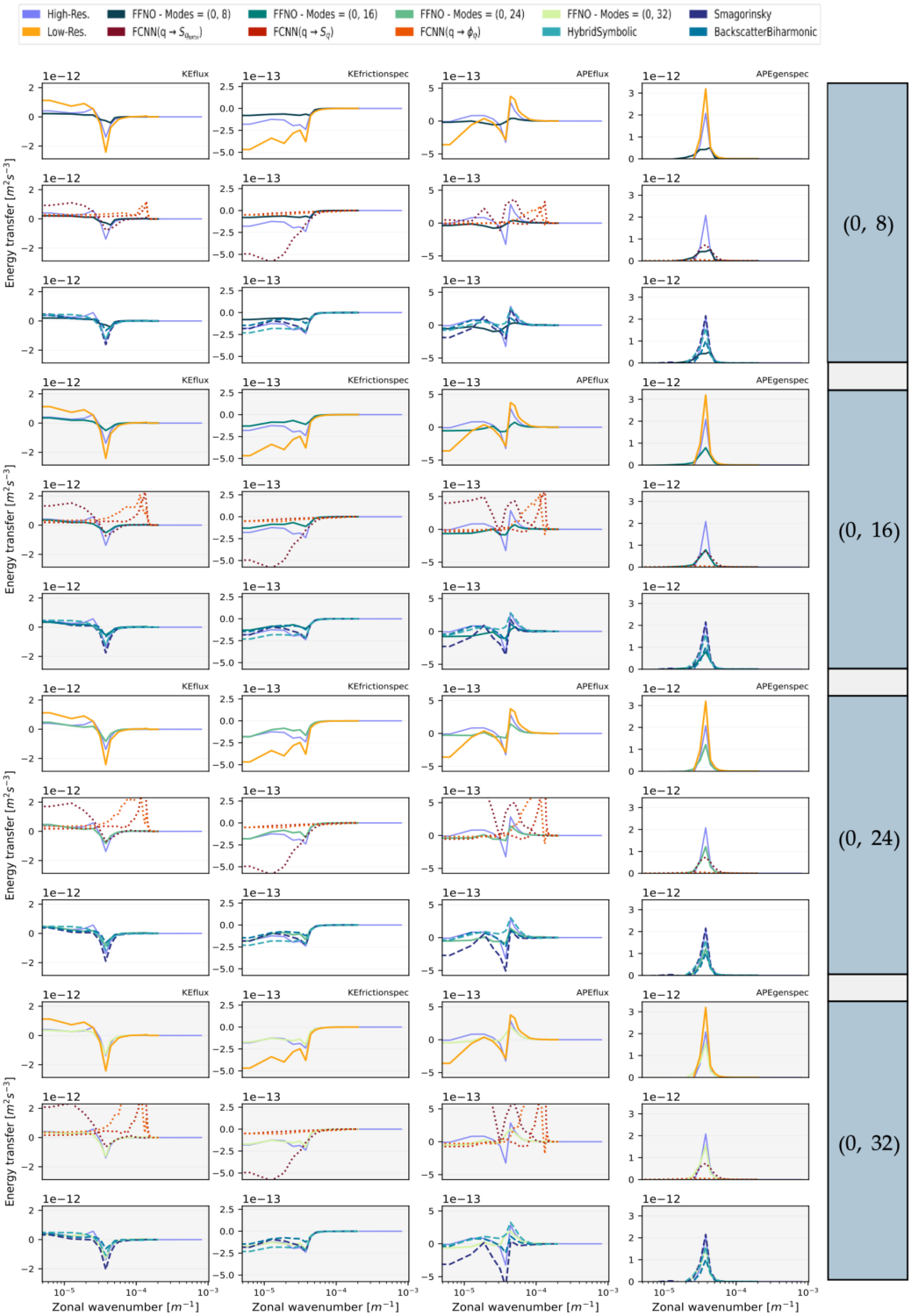


Figure 46: | **Online - Phase 6 - Energy budget** | This table displays energy spectra for **KEflux**, **KEfrictionspec**, **APEflux**, and **APEgenspec** using parameterizations of Tab.7, these were trained on **full 5000** and tested on **jets online**. Each parameterization spectrum is compared against high-resolution and various low-resolution simulations, including neural networks from Ross et al., 2023 and analytical parameterizations from Smagorinsky, 1963; Jansen and Held, 2014; Zanna and Bolton, 2020.

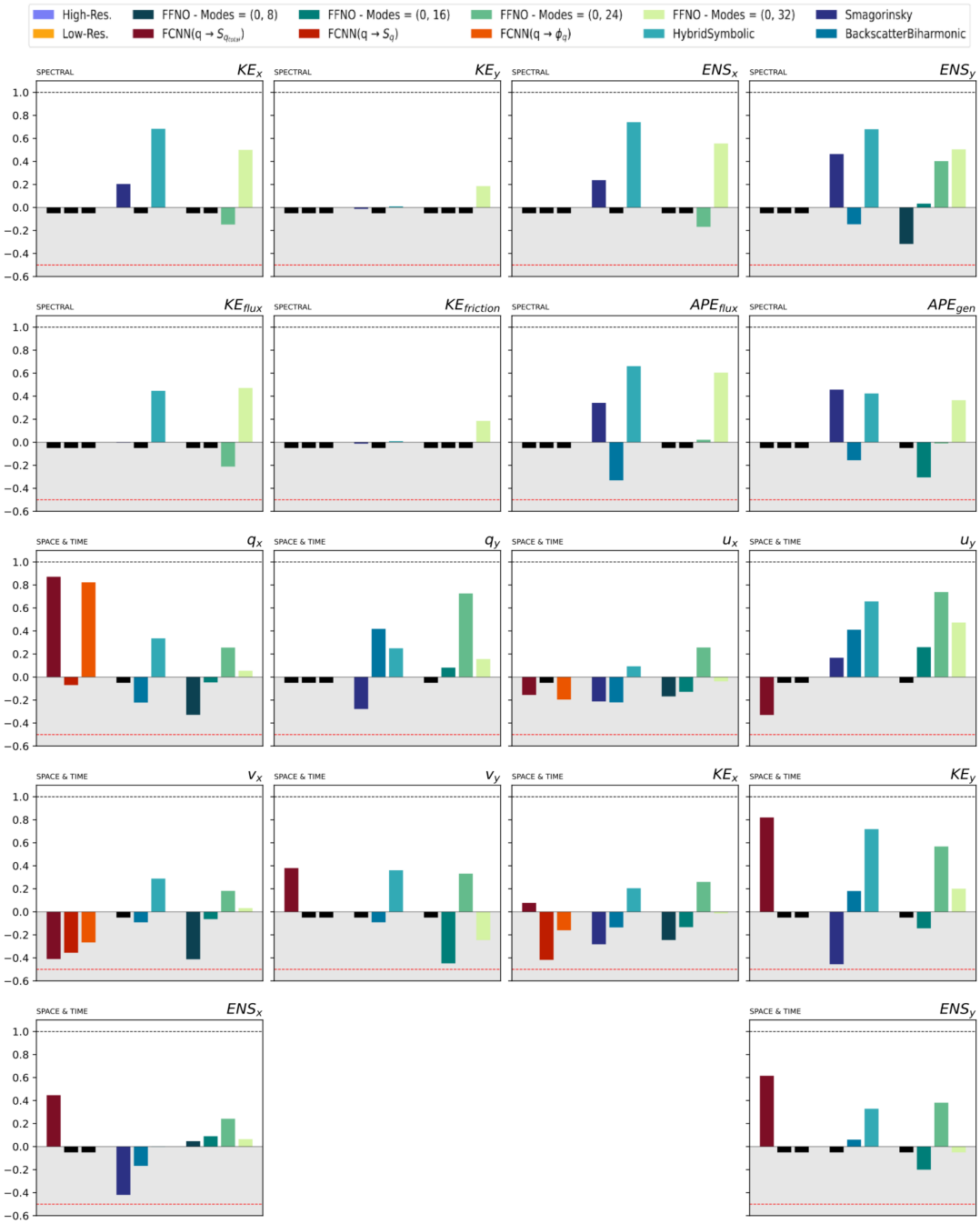


Figure 47: | **Online - Phase 6 - Similarities** | This table provides a summary of the Earth mover's distance, reformulated as a similarity metric for various flow quantities represented in either spectral or spatiotemporal domains. A value approaching 1 indicates strong agreement between the distribution obtained from high-resolution simulations and the current observations. Negative values are considered unfavorable, and values lower than -0.5 are disregarded. The tested parameterizations comes from Tab.7, they are trained on **full 5000** and tested on **jets online**. For comparison, the results of neural networks (Ross et al., 2023) and analytical parameterizations are also presented (Smagorinsky, 1963; Jansen and Held, 2014; Zanna and Bolton, 2020).

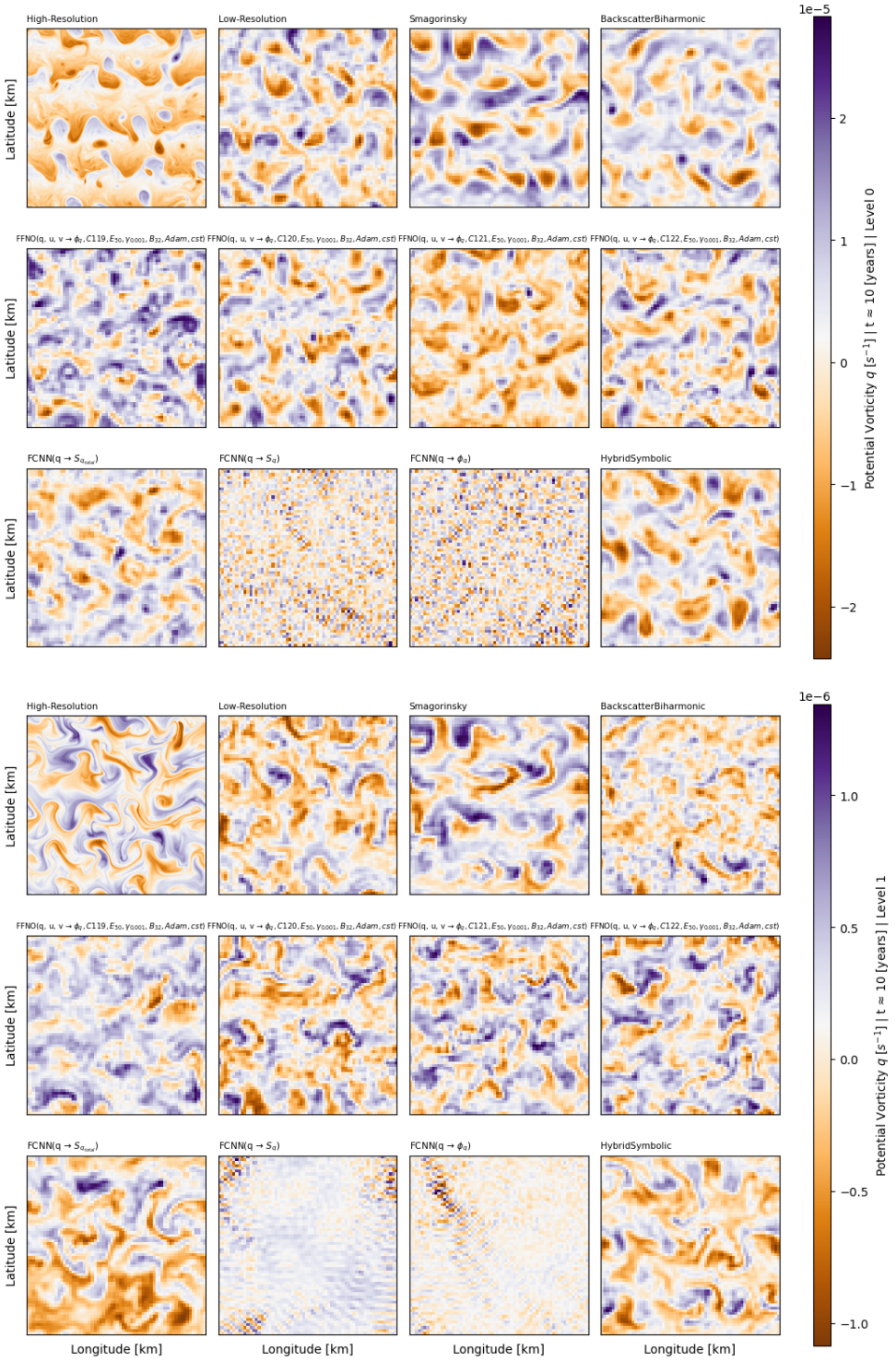


Figure 48: | **Online - Phase 6 - Potential vorticity** | Visualization of potential vorticity is presented for both upper (first three rows) and lower (last three rows) layers across different simulation types, indicated at the top of each image. Each image represents the q value spanning the entire computational domain after 10 years of simulations. The objective is to emphasize and visualize simulations that lose their physical relevance, becoming mere pixel grids, and to illustrate the divergence from the high-resolution simulation. Furthermore, the evaluated parameterizations are detailed in Tab.7, they were trained using **full 5000** and assessed against **jets online**.