A posteriori learning for quasi-geostrophic turbulence parametrization

Hugo Frezat, Julien Le Sommer, Ronan Fablet, Guillaume Balarac, Redouane Lguensat

JAMES | Journal of Advances in Modeling Earth Systems

RESEARCH ARTICLE
10.1029/2022MS003124

Special Section:
Machine learning application to Earth system modeling

A Posteriori Learning for Quasi-Geostrophic Turbulence Parametrization

Hugo Frezat[1,2,3], Julien Le Sommer[2], Ronan Fablet[3], Guillaume Balarac[1,4], and Redouane Lguensat[5]

[1]University Grenoble Alpes, CNRS UMR LEGI, Grenoble, France, [2]University Grenoble Alpes, CNRS UMR IGE, Grenoble, France, [3]IMT Atlantique, CNRS UMR Lab-STICC, INRIA Team Odyssey, Brest, France, [4]Institut Universitaire de France (IUF), Paris, France, [5]Institut Pierre Simon Laplace, IRD, Sorbonne Université, Paris, France

Key Points:

- Subgrid parametrizations can be learned end-to-end with a posteriori criteria involving model integration over several time-steps
- A posteriori learning for quasi-geostrophic turbulent flows can solve numerical stability issues related to energy backscatter
- Learned parametrizations outperform existing baselines for various evaluation metrics and apply to different flow configurations

Correspondence to:
H. Frezat,
hugo.frezat@univ-grenoble-alpes.fr

Citation:
Frezat, H., Le Sommer, J., Fablet, R., Balarac, G., & Lguensat, R. (2022). A posteriori learning for quasi-geostrophic turbulence parametrization. *Journal of Advances in Modeling Earth Systems*, *14*, e2022MS003124. https://doi.org/10.1029/2022MS003124

Received 7 APR 2022
Accepted 27 SEP 2022

Author Contributions:
Conceptualization: Hugo Frezat, Julien Le Sommer, Ronan Fablet, Guillaume Balarac
Data curation: Hugo Frezat
Funding acquisition: Julien Le Sommer, Ronan Fablet, Guillaume Balarac
Investigation: Hugo Frezat, Julien Le Sommer, Ronan Fablet, Guillaume Balarac
Methodology: Hugo Frezat

Abstract The use of machine learning to build subgrid parametrizations for climate models is receiving growing attention. State-of-the-art strategies address the problem as a supervised learning task and optimize algorithms that predict subgrid fluxes based on information from coarse resolution models. In practice, training data are generated from higher resolution numerical simulations transformed in order to mimic coarse resolution simulations. By essence, these strategies optimize subgrid parametrizations to meet so-called a priori criteria. But the actual purpose of a subgrid parametrization is to obtain good performance in terms of a posteriori metrics which imply computing entire model trajectories. In this paper, we focus on the representation of energy backscatter in two-dimensional quasi-geostrophic turbulence and compare parametrizations obtained with different learning strategies at fixed computational complexity. We show that strategies based on a priori criteria yield parametrizations that tend to be unstable in direct simulations and describe how subgrid parametrizations can alternatively be trained end-to-end in order to meet a posteriori criteria. We illustrate that end-to-end learning strategies yield parametrizations that outperform known empirical and data-driven schemes in terms of performance, stability, and ability to apply to different flow configurations. These results support the relevance of differentiable programming paradigms for climate models in the future.

Plain Language Summary Climate projection and weather forecast heavily rely on computer simulations. But, if the physical laws governing the evolution of the climate system are well known, their simulation is still rather challenging. Fluid flows being essentially turbulent, small details at fine scales can have a tremendous impact on larger scales. Still, because of the limitations in computing power, all these interactions across scales cannot be explicitly resolved in computer simulations. Some of these interactions can only be represented approximately, and the design of these approximations is an active research area. Here, we describe a new method which leverages recent advances in machine learning. We propose to train an approximate representation of unresolved scales of motions that optimizes the quality of the climate model over some temporal horizon. This results in more accurate and stable predictions. Our method shows very promising results in toy example flow simulations, but its deployment at scale may seriously challenge the overall design of legacy climate models.

1. Introduction

The representation of unresolved processes is a key source of uncertainty in weather and climate models. Climate science and weather forecasting indeed heavily rely on numerical simulations of the Earth's atmosphere and oceans (Bauer et al., 2015; Neumann et al., 2019). But even the most advanced applications are currently far from resolving explicitly the wide variety of space-time scales and physical processes involved. This will likely remain the case for the foreseeable future because of the nonlinearity of fluid dynamics and thermodynamics, and because of the finite nature of computational resources (Fox-Kemper et al., 2014; Schneider, Teixeira, et al., 2017). Weather and climate models will therefore keep relying on approximated representations of the effect of unresolved processes in the form of subgrid *parametrization* schemes (Fox-Kemper et al., 2019; Schneider, Lan, et al., 2017). Parametrization schemes accounting for the impact of turbulence in the atmosphere and oceans at various scales will in particular remain essential components of these models.

Project Administration: Julien Le Sommer, Ronan Fablet, Guillaume Balarac
Resources: Hugo Frezat
Software: Hugo Frezat
Supervision: Julien Le Sommer, Ronan Fablet, Guillaume Balarac
Validation: Hugo Frezat, Redouane Lguensat
Visualization: Hugo Frezat
Writing – original draft: Hugo Frezat
Writing – review & editing: Hugo Frezat, Julien Le Sommer, Ronan Fablet, Guillaume Balarac, Redouane Lguensat

Parametrizations of unresolved turbulent motions are usually based on first principles, physics, idealized experiments, field observations, and high-resolution simulations. Their design involves a mixture of empirical and process-based modeling. Process-based models are formulated, tested and calibrated with experiments performed in the field, in the laboratory or with computers (Stensrud, 2009). On this basis, the actual parametrization scheme estimates a tendency term for the target model from its resolved variables. The underlying conceptual framework can rely on some ensemble averaging procedure, so that the parametrization intends to capture the bulk statistical effect of unresolved processes, as for instance for turbulence models (Mellor, 1985). Alternatively, one may consider a spatial filtering procedure and the parametrization can then exploit the scale-invariant properties of turbulence, as in Large Eddy Simulation models (Lesieur et al., 2005). This latter framework is used for instance for the parametrizations of ocean macro-turbulence in eddy-rich ocean models (Fox-Kemper & Menemenlis, 2008).

Recently, the use of machine learning (ML) for better parametrizing unresolved processes in weather and climate models has gained momentum. Calibrating physics-based parametrization schemes against observation with ML and emulators is for instance becoming common practice (Couvreux et al., 2021; Ollinaho et al., 2013; Schneider, Lan, et al., 2017). Emulation approaches based on ML have also been proposed as a strategy for accelerating or regularizing existing schemes (Chantry et al., 2021; Meyer et al., 2021; Ukkonen et al., 2020). ML also provides new means to design new subgrid parametrization schemes from high-fidelity simulations. In this context, ML may learn a mapping which predicts the tendency term due to unresolved subgrid effects from resolved quantities available in a target model. In atmospheric models, these approaches have been used to improve the representation of cloud micro-physics and moist processes (Brenowitz & Bretherton, 2018; Krasnopolsky et al., 2013; O'Gorman & Dwyer, 2018; Rasp et al., 2018; Seifert & Rasp, 2020). In ocean models, it is expected that the representation of macro-turbulence could be improved with similar approaches (Bolton & Zanna, 2019; Guillaumin & Zanna, 2021).

The design of parametrizations with ML builds on the rise of scientific machine learning (SciML) and its broad application to physical sciences. SciML is an emerging field, which bridges scientific computing and ML. Some recent key developments in this field have been motivated both *from* physical insights and *for* their applications to physical sciences, especially in fluid dynamics (Carleo et al., 2019; Thuerey et al., 2021). The conceptual developments in ML motivated by applications to problems governed by partial differential equations (Long et al., 2018; Raissi et al., 2019; Sirignano & Spiliopoulos, 2018) have for instance gradually freed ML from its black-box reputation. The design of parametrization schemes now directly benefits from ML approaches for dynamical system identification and equation discovery (Brunton et al., 2016; Zanna & Bolton, 2020). The ability to embed symmetries and law invariances into neural networks (Alet et al., 2021; Cohen & Welling, 2016; Cranmer et al., 2020) will also likely be important in the design of parametrization schemes (Frezat et al., 2021), and in applications of ML to fluid mechanics in general (Brunton et al., 2020; Vinuesa & Brunton, 2021).

But ML-based approaches to subgrid parametrizations are still mostly based on a priori learning strategies, which could limit their performance and applicability. There are indeed two different sorts of evaluation metrics for measuring the precision of subgrid models in turbulent simulations (Pope, 2000). A priori metrics, on the one hand, measure to what extent a given subgrid model is able to predict a tendency term due to unresolved subgrid effects at a fixed time. A posteriori metrics, on the other hand, require to perform simulations with the subgrid model, and measure its integrated impact on the simulated flows. The common strategy for learning subgrid parametrizations is to formulate a supervised learning task from a high-resolution reference simulation data set. In practice, learned parametrizations result from the minimization of a cost function based on some a priori metrics measuring how well a mapping can predict unresolved fluxes from coarse-grain quantities. Still, with such strategy, what we *really* intend to optimize is the ability of the parametrization to yield *good* solutions, when used a posteriori in numerical simulations. In principle, the versatility of ML algorithms should allow us to train parametrization schemes with learning criteria based on a posteriori metrics, adopting the so-called *end-to-end* learning framework (Glasmachers, 2017). But surprisingly, there are very few published examples of end-to-end learning strategies in computational fluid dynamics (Kochkov et al., 2021; Sirignano et al., 2020; Stachenfeld et al., 2021). It is therefore yet unclear how subgrid parametrizations trained with a priori and a posteriori compare in terms of performance, stability, and ability to apply to different flow conditions.

Flows governed by quasi-geostrophic (QG) dynamics provide an interesting and challenging testbed to evaluate learning strategies for subgrid parametrizations. QG theory indeed proposes a simple framework for studying

geophysical flows constrained by earth rotation and stratification, as for instance large-scale atmospheric dynamics and ocean macro-turbulence (Cushman-Roisin & Beckers, 2011). The two-dimensional turbulence emerging from barotropic QG dynamics (Boffetta & Ecke, 2012; Majda & Wang, 2006) exhibits a dual cascade scenario with inverse energy transfers to larger scales and direct enstrophy transfers to smaller scales (Kraichnan, 1967; Thuburn et al., 2014). Because of this inverse energy cascade, developing subgrid parametrizations for QG flows is a challenging task, as the stability of numerical integration schemes is directly controlled by the rate of energy backscatter, that is, transfers from subgrid to resolved scales (Carati et al., 1995; Lilly, 1992). As a consequence, a large number of subgrid parametrization schemes have been proposed for two-dimensional turbulence (see e.g., Danilov et al., 2019 for a review) and well documented flow configurations with performance metrics for parametrization are readily available (Graham & Ringler, 2013). Unsurprisingly, attempts to learn subgrid parametrizations for two-dimensional turbulence with ML have been less successful than for other types of turbulent flows (Guan, Chattopadhyay, et al., 2022; Maulik et al., 2019). In particular, ad-hoc solutions had to be implemented in order to ensure the numerical stability of the learned schemes under energy backscatter conditions. For instance, Maulik et al. (2019) use a clipping post-processing procedure to remove negative diffusivity while Guan, Chattopadhyay, et al. (2022) mitigate this problem in decaying turbulence by increasing the size of the training data set. More recently, Guan, Subel, et al. (2022) and Pawar et al. (2022) demonstrated how incorporating physics in the models could lead to stable simulations that requires less data for training and generalizes better. Up to now, however, none of the published works investigates the long-term statistical performance of learned schemes far beyond the decorrelation horizon. Learning stable parametrizations for two-dimensional turbulence in QG flows is therefore still an open problem.

In this work, we compare parametrizations for two-dimensional turbulence obtained with different learning strategies, at fixed computational complexity. In particular, we show that we are able to train a model based on a posteriori metrics with an end-to-end learning strategy. Through evaluation on three different configurations (decay, wind-forcing and beta-effect), the end-to-end learning strategy is shown to yield stable parametrizations that outperform previous physics-based and NN-based models without any explicit postprocessing step. Statistical metrics on long-term spectral transfers are shown to be in excellent agreement to direct numerical simulations (DNS), which is particularly encouraging for future climate models. The paper is organized as follows: In Section 2, we present the a priori and a posteriori learning strategies and the type of metrics they are respectively able to optimize. The application to QG parametrizations is described in Section 3 with the numerical setup and baselines used in the evaluation. Results are presented both for short-term and long-term statistics for three different configurations in Section 4. Finally, we discuss the limitations and implications of the described strategies for realistic large-scale solvers.

2. Learning Strategies

In this study, we address the simulation of the time evolution of geophysical quantities $\mathbf{y}(t)$. We assume the underlying governing equations to be known. Let us denote by $f(\mathbf{y})$ these true dynamics. The numerical integration of this system being either impossible or expensive, we aim at solving the time evolution of reduced variables $\bar{\mathbf{y}}(t)$ such that:

$$\begin{cases} \dfrac{\partial \mathbf{y}}{\partial t} = f(\mathbf{y}), & \mathbf{y} \in \Omega \\ \dfrac{\partial \bar{\mathbf{y}}}{\partial t} = g(\bar{\mathbf{y}}) + \mathcal{M}(\bar{\mathbf{y}}), & \bar{\mathbf{y}} \in \bar{\Omega} \\ \mathcal{T}(\mathbf{y}) = \bar{\mathbf{y}} \end{cases} \tag{1}$$

where $\bar{\Omega} \subset \Omega$, g a reduced-order operator, $\mathcal{M}$ a subgrid-scale (SGS) parametrization and $\mathcal{T}$ is a projection operator that maps true variables to reduced ones. The objective in reduced-order modeling is to design operator g such that the evolution of the reduced variables matches the projection $\mathcal{T}(\mathbf{y})$ of the true variables $\mathbf{y}$. We note that for some reduced order problems, we identify $f = g$ with variables existing on different spaces or dimensionalities.

Within a learning framework, one states the identification of SGS term $R(\mathbf{y}) = \mathcal{T}(f(\mathbf{y})) - g(\mathcal{T}(\mathbf{y}))$ as a learning problem from reduced variables for a parametrization $\mathcal{M}(\bar{\mathbf{y}}|\theta)$ where θ are trainable model parameters. Under the assumption that projection operator $\mathcal{T}$ commutes with partial derivatives, the most classic approach comes

to train parametrization $\mathcal{M}(\bar{\mathbf{y}}|\theta)$ as a functional approximation of closure term R. This approach has been widely explored in the recent literature (Bolton & Zanna, 2019; Vollant et al., 2017). It does not however constrain the trained parametrization to behave as expected when implemented in the solver of the reduced-order system. In this respect, an end-to-end framework would appear as an appealing approach to explicitly state the SGS parametrization problem according to the best possible approximation of the true reduced variables. Such end-to-end approaches have shown many advantages in the approximation of differential equation in general (Bakarji & Tartakovsky, 2021; Chen et al., 2018; Fablet et al., 2021). When applied to physical problems, they are often referred as differentiable physics (de Avila Belbute-Peres et al., 2018; Holl et al., 2020; Um et al., 2020), since they require the gradient of all the considered operators and solvers to be available for the optimization algorithm. Overall, these two categories of learning approaches differ in the space where the training is performed, similarly to the definition of a priori and a posteriori metrics (Pope, 2000) for the benchmarking of SGS parametrizations. This is the reason why we refer to a priori and a posteriori learning strategies as detailed in the subsequent.

2.1. A Priori Learning

The a priori learning strategy comes to learn SGS parametrization using training metrics defined on instantaneous quantities, that is, a direct measure of the accuracy of the model based on the predicted SGS term $R(\mathbf{y})$. The a priori loss $\mathcal{L}_{\text{prio}}$ has the form,

$$\mathcal{L}_{\text{prio}}(\mathcal{M}) := \ell\left(R(\mathbf{y}), \mathcal{M}\left(\bar{\mathbf{y}}|\theta\right)\right) \tag{2}$$

where $\mathcal{M}$ is a given SGS model to be evaluated. The most common a priori metrics ℓ found in the fluid dynamics community are the mean squared error (MSE) and the correlation between true and predicted SGS terms. Training a NN-based parametrization according to a priori setting then comes to building a representative ground-truth data set $\{R(\mathbf{y}_i), \bar{\mathbf{y}}_i\}_n$ of paired SGS terms and reduced variables to solve the following minimization problem with respect to model parameters θ

$$\underset{\theta}{\operatorname{argmin}}\, \mathcal{L}_{\text{prio}}(\mathcal{M}) \equiv \underset{\theta}{\operatorname{argmin}}\, \ell\left(\{R(\mathbf{y}_i)\}, \mathcal{M}\left(\{\bar{\mathbf{y}}_i\}|\theta\right)\right). \tag{3}$$

Solving for (Equation 3) requires evaluation of the partial derivative of the a priori loss $\mathcal{L}_{\text{prio}}$ with respect to parameter θ, which only involves the gradient of $\mathcal{M}$,

$$\frac{\partial \mathcal{L}_{\text{prio}}}{\partial \theta} = \frac{\partial \mathcal{L}_{\text{prio}}}{\partial R}\frac{\partial R}{\partial \theta} + \frac{\partial \mathcal{L}_{\text{prio}}}{\partial \mathcal{M}}\frac{\partial \mathcal{M}}{\partial \theta} = \frac{\partial \mathcal{L}_{\text{prio}}}{\partial \mathcal{M}}\frac{\partial \mathcal{M}}{\partial \theta}. \tag{4}$$

This approach has been applied to scalar (Frezat et al., 2021; Portwood et al., 2021; Vollant et al., 2017) and momentum (Beck et al., 2019; Gamahara & Hattori, 2017; Xie et al., 2020; Yuan et al., 2020) parametrizations of three-dimensional turbulence on different configurations. The two-dimensional case is also well documented in decaying (Guan, Chattopadhyay, et al., 2022; Maulik et al., 2019; Pawar et al., 2020) and double-gyre (Bolton & Zanna, 2019; Zanna & Bolton, 2020) configurations. We may emphasize that, by construction, the a priori learning strategy shall lead to the best a priori results, which shall translate into a good instantaneous prediction of the SGS term according to metrics $\mathcal{L}_{\text{prio}}$.

2.2. A Posteriori Learning

The a posteriori learning strategy states the SGS parametrization problem as the approximation of the true reduced variables according to some a posteriori metrics. This is important since it is possible for a model to perform well a priori while failing a posteriori, the most common factor being numerical instabilities due to the lack of small-scale energy dissipation (Guan, Chattopadhyay, et al., 2022; Maulik et al., 2019). Let us denote by Φ the flow operator that advances the reduced system in time, that is,

$$\Phi_\theta^{t_1}\left(\bar{\mathbf{y}}\left(t_0\right)\right) = \bar{\mathbf{y}}\left(t_0\right) + \int_{t_0}^{t_1} g\left(\bar{\mathbf{y}}(t)\right) + \mathcal{M}\left(\bar{\mathbf{y}}(t)|\theta\right) \mathrm{d}t. \tag{5}$$

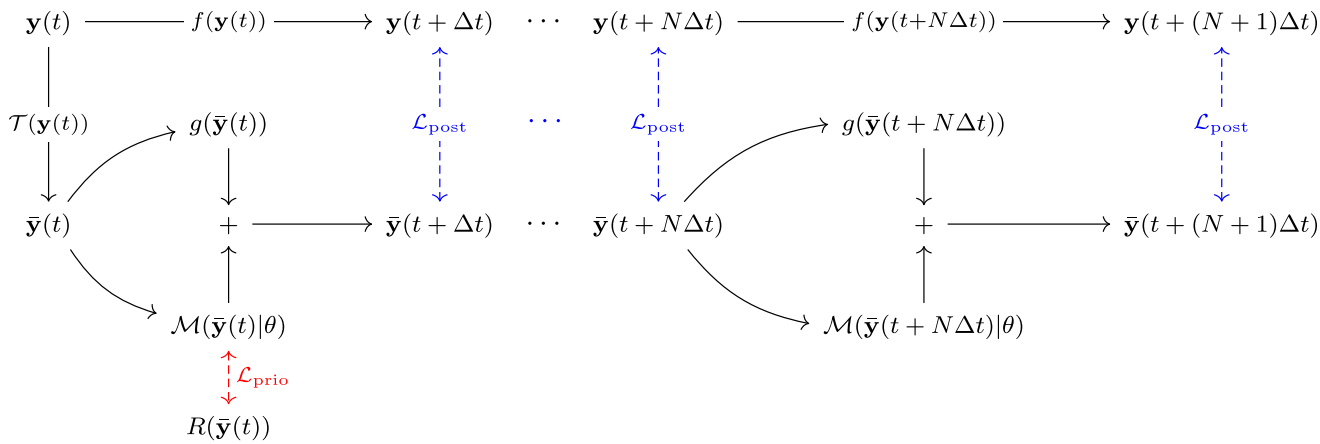

Figure 1. Sketch of one learning step for the a priori and a posteriori strategies. The a priori loss is computed at instantaneous time t (dashed, red), while the a posteriori loss involves several states forward in time (dashed, blue).

Numerically speaking, flow operator Φ involves a time integration scheme (see Figure 1) from start to end time t_0 and t_1, respectively. Following recent advances in neural integration schemes (Chen et al., 2018; Ouala et al., 2021), we may consider here both explicit and adaptive schemes. Let $\ell(\mathbf{y}(t), \bar{\mathbf{y}}(t))$ be some a posteriori metrics defined in the true $\{\mathbf{y}(t)\}_{t\in[0,T]}$ and reduced $\{\bar{\mathbf{y}}(t)\}_{t\in[0,T]}$ spaces and a representative data set of trajectories over a time interval $[0, T]$. The a posteriori learning strategy comes to minimize an a posteriori training loss,

$$\mathcal{L}_{\text{post}} := \ell\left(\{\mathbf{y}(t)\}_{t\in[t_0,t_1]}, \left\{\Phi_\theta^t(\bar{\mathbf{y}}(t_0))\right\}_{t\in[t_0,t_1]}\right) \tag{6}$$

Now, the a posteriori minimization problem involves the time integration of Φ on sub-intervals $[t_0, t_1]$ from the data set of trajectories spanning the interval $[0, T]$, with initial reduced states typically taken to be $\bar{\mathbf{y}}(t_0) = \mathcal{T}(\mathbf{y}(t_0))$,

$$\underset{\theta}{\text{argmin}}\mathcal{L}_{\text{post}} \equiv \underset{\theta}{\text{argmin}}\ell\left(\{\mathbf{y}(t)\}_{t\in[t_0,t_1]}, \left\{\Phi_\theta^t(\bar{\mathbf{y}}(t_0))\right\}_{t\in[t_0,t_1]}\right), \forall [t_0, t_1] \in [0, T]. \tag{7}$$

Then from the Leibniz integral rule, updating model parameters θ requires the flow partial derivative, that is,

$$\frac{\partial\mathcal{L}_{\text{post}}}{\partial\theta} = \frac{\partial\mathcal{L}_{\text{post}}}{\partial\mathbf{y}(t_1)}\frac{\partial\mathbf{y}(t_1)}{\partial\theta} + \frac{\partial\mathcal{L}_{\text{post}}}{\partial\Phi}\frac{\partial\Phi}{\partial\theta} = \frac{\partial\mathcal{L}_{\text{post}}}{\partial\Phi}\left(\bar{\mathbf{y}}(t_0) + \int_{t_0}^{t_1}\frac{\partial g(\bar{\mathbf{y}}(t))}{\partial\theta} + \frac{\partial\mathcal{M}(\bar{\mathbf{y}}(t)|\theta)}{\partial\theta}\mathrm{d}t\right). \tag{8}$$

This equation makes explicit that the gradient-based optimization of the a posteriori criterion involves the computation of the gradient with respect to all the components of the forward model, that is, dynamical operator g as well as the considered time integration scheme. Assuming that one can run all components within a differentiable programming framework (here, PyTorch [Paszke et al., 2019]), the embedded automatic differentiation (AD) tools make these computations direct with no additional programming cost. In our experiments, this comes to performing an AD for an explicit fourth-order Runge-Kutta scheme with N discrete time-steps, which defines the temporal horizon $T = N\Delta t$.

The a posteriori strategy significantly widens the range of metrics which can be considered to calibrate the SGS parametrization. We illustrate this modeling flexibility for QG turbulence in the next section. We may point out that this a posteriori learning strategy has recently been explored for temporally developing plane turbulent jets (MacArt et al., 2021) and the short-term simulation of short-term two-dimensional flows (Kochkov et al., 2021). Here, we explore further its relevance for two-dimensional geophysical flows, including a benchmarking with the a priori setting for different flow configurations.

3. Application to QG Turbulence

Geophysical turbulence is widely acknowledged to involve energy backscatter. This makes SGS parametrization a key issue for the simulation of ocean and atmosphere dynamics (Graham & Ringler, 2013; Jansen et al., 2015; Juricke et al., 2020). As case-study framework, we consider QG flows. While providing an approximate yet representative model for rotating stratified flows found in atmosphere and ocean dynamics, they involve relatively complex SGS features that make the learning problem non trivial. As such, QG flows are regarded as an ideal

playground to explore and assess the relevance of a priori and a posteriori learning strategies for SGS parametrization in geophysical turbulence. QG equations (Majda & Wang, 2006) are given by,

$$\partial_t \omega + J(\psi, \omega) = \nu \nabla^2 \omega - \mu\omega - \beta \partial_x \psi + F \tag{9}$$

which is equivalent to the transport of vorticity ω, obtained by taking the curl of the incompressible Navier-Stokes equations, that is, $\nabla \cdot \mathbf{u} = 0$ and $\omega = \nabla \times \mathbf{u}$ and applying beta-plane approximation, hydrostatic and geostrophic balances. In addition, we have,

$$\mathbf{u} = (-\partial_y \psi, \partial_x \psi) \tag{10}$$

$$\omega = \nabla^2 \psi \tag{11}$$

where ψ is the stream function, $\mathbf{u}$ the velocity and $J(\psi, \omega) = \partial_x \psi \partial_y \omega - \partial_y \psi \partial_x \omega$ is the nonlinear Jacobian operator. The model is parametrized by viscosity ν, linear drag coefficient μ, Rossby parameter β and a source term F. The QG equations have two invariants in the limit of inviscid flow (Bouchet & Venaille, 2012), for energy

$$E = \frac{1}{2} \int \mathbf{u}^2 \mathrm{d}r \tag{12}$$

and enstrophy

$$Z = \frac{1}{2} \int \omega^2 \mathrm{d}r. \tag{13}$$

In order to study scales interactions, we introduce the enstrophy spectrum, $Z(k)$ in spectral space as the enstrophy contained in the shell of radius k,

$$Z(k) = \frac{1}{2} \int_{|\mathbf{k}|=k} \hat{\omega}(\mathbf{k}) \hat{\omega}^*(\mathbf{k}) \mathrm{d}S(\mathbf{k}) \tag{14}$$

where the Fourier transform is represented by $\hat{\cdot}$, complex conjugation by $\cdot^*$ and with $\mathrm{d}S(\mathbf{k})$ the integration element of the shell of radius k from wavevector $\mathbf{k}$. The evolution equation of $Z(k)$ writes in spectral space,

$$\frac{d}{dt} Z(k) = -D(k) - Q(k) - B(k) + F(k) + T(k) \tag{15}$$

where the different terms of the right hand side are related to various effects: external energy source F, dissipation D, large-scale drag Q, beta-plane effect B and transfer between scales T. This last term writes

$$T(k) = \int_{|\mathbf{k}|=k} \Re \left\{ \hat{\omega}^*(\mathbf{k}) \hat{J}(\psi, \omega)(\mathbf{k}) \right\} \mathrm{d}S(\mathbf{k}). \tag{16}$$

This allows to define the enstrophy flux (Gupta et al., 2019) through the wavenumber k as

$$\Pi_Z(k) = -\int_0^k T\left(k'\right) \mathrm{d}k' \tag{17}$$

which is a key quantity to measure effect of SGS modeling on enstrophy distribution on the range of resolved scales.

3.1. SGS Parametrization for QG Dynamics

The derivation of the reduced model for QG dynamics follows the same procedure that is described for fluid dynamics in general. Assuming a known projection operator $\mathcal{T}$ from Equation 1 at spatial coordinate $\mathbf{x}$ given as a discretization $\mathcal{D} : \Omega \to \bar{\Omega}$ and the convolution of $\mathbf{y}$ with a kernel function $G(\mathbf{x})$ (Leonard, 1975),

$$\bar{\mathbf{y}}(\mathbf{x}) = \mathcal{T}(\mathbf{y}(\mathbf{x})) = \mathcal{D}\left[\int G\left(\mathbf{x} - \mathbf{x}'\right) \mathbf{y}\left(\mathbf{x}'\right) \mathrm{d}\mathbf{x}' \right]. \tag{18}$$

We can then derive the equations which govern the evolution of reduced vorticity $\bar{\omega}$ as,

$$\begin{cases} \partial_t \omega + J(\psi, \omega) = \nu \nabla^2 \omega - \mu \omega - \beta \partial_x \psi + F, & \omega \in \Omega \\ \partial_t \bar{\omega} + J(\bar{\psi}, \bar{\omega}) = \nu \nabla^2 \bar{\omega} - \mu \bar{\omega} - \beta \partial_x \bar{\psi} + \bar{F} + \underbrace{J(\bar{\psi}, \bar{\omega}) - \overline{J(\psi, \omega)}}_{R(\psi,\omega)}, & \bar{\omega} \in \bar{\Omega} \end{cases} \tag{19}$$

where $R(\psi, \omega)$ is the SGS term. For convenience, note that the reduced term can be expressed in a flux formulation,

$$R(\psi, \omega) = \nabla \cdot \left(\bar{\mathbf{u}} \bar{\omega} - \overline{\mathbf{u}\omega} \right). \tag{20}$$

In this context, $\bar{\omega}$ is only solved for the largest scales of the flow, and $R(\psi, \omega)$ accounts for the effect of unresolved motions on the resolved scales. The SGS term is thus not known from the reduced variables because of the nonlinear interactions of small-scale dynamics $\overline{J(\psi, \omega)}$. Following the notations introduced in Section 2, we aim to identify a QG SGS parametrization $\mathcal{M}(\bar{\psi}, \bar{\omega}|\theta)$ given the parametrization for operator g by Equation 19 using both a priori and a posteriori learning strategies.

In order to study the effect of the SGS parametrization on scales interactions, we will consider the enstrophy spectrum $Z(k)$ and the associated enstrophy flux $\Pi_Z(k)$ in various flow configurations. Note that when the governing equation of reduced vorticity $\bar{\omega}$ is solved, the transfer term is now split in a resolved and a modeled part, as

$$T(k) = \int_{|\mathbf{k}|=k} \Re \left\{ \underbrace{\hat{\bar{\omega}}^*(\mathbf{k}) \hat{J}(\bar{\psi}, \bar{\omega})(\mathbf{k})}_{\text{resolved}} - \underbrace{\hat{\bar{\omega}}^*(\mathbf{k}) \hat{R}(\psi, \omega)(\mathbf{k})}_{\text{modeled}} \right\} \mathrm{d}S(\mathbf{k}). \tag{21}$$

We can also look at the resolved enstrophy equilibrium in physical space. The governing equation for $\bar{z} = \frac{1}{2}\bar{\omega}^2$ writes

$$\partial_t \bar{z} = \nu \bar{\omega} \nabla^2 \bar{\omega} - \mu \bar{\omega} \bar{\omega} - \beta \bar{\omega} \partial_x \bar{\psi} + \bar{\omega} \bar{F} - \bar{\omega} J(\bar{\psi}, \bar{\omega}) + \underbrace{\bar{\omega} R(\psi, \omega)}_{T_{\bar{z}}} \tag{22}$$

where the last term $T_{\bar{z}}$ shows the direct effect of the SGS term on the resolved enstrophy equilibrium. From Equation 20, this last term can be expressed as,

$$T_{\bar{z}} = \nabla \cdot \left(\bar{\omega} \left(\bar{\mathbf{u}} \bar{\omega} - \overline{\mathbf{u}\omega} \right) \right) - \left(\bar{\mathbf{u}} \bar{\omega} - \overline{\mathbf{u}\omega} \right) \cdot \nabla \bar{\omega}, \tag{23}$$

with first term related to diffusion, and second term related to the transfer between resolved and unresolved scales: if $\left(\bar{\mathbf{u}} \bar{\omega} - \overline{\mathbf{u}\omega} \right) \cdot \nabla \bar{\omega} > 0$, there is a direct transfer from resolved to unresolved scales (forwardscatter), but if $\left(\bar{\mathbf{u}} \bar{\omega} - \overline{\mathbf{u}\omega} \right) \cdot \nabla \bar{\omega} < 0$, there is a transfer from unresolved to resolved scale (backscatter). The latter will be the key quantity to study the ability of models to take into account backscatter. However, because models are built directly for $R(\psi, \omega)$ the above decomposition can not be performed, and only $T_{\bar{z}}$ can be considered.

3.2. Numerical Solver

Equation 9 is solved using a pseudo-spectral differentiable code with full 3/2 dealiasing (Canuto et al., 2012) and a classical fourth-order Runge-Kutta time integration scheme. The system is defined in a squared domain $\Omega \in [-\pi, \pi]^2$ with a Fourier basis, that is, double-periodic boundary conditions $\partial\Omega$ on $N_{\text{true}} = (N_x, N_y)$ grid points with uniform spacing $\Delta_{\text{true}} = \Omega N_{\text{true}}^{-1}$. The reduced states are obtained by projecting the true (or high-resolution) states through a convolution with a spatial kernel $G_\delta(k)$ at spatial scale $\delta > 0$ followed by a discretization on the reduced grid $\bar{\Omega}$, that is, with larger spacing $\Delta_{\text{reduced}} = \delta \Delta_{\text{true}}$ equivalent to a sharp cutoff,

$$\bar{\omega}(k) := (\omega * G_\delta) \left(|k| < \pi \Delta_{\text{reduced}}^{-1} \right). \tag{24}$$

It has been shown previously that SGS parametrizations can perform differently depending on the type of filter used in the evaluations (Piomelli et al., 1988; Zhou et al., 2019). We then aim to evaluate how the choice of the filter affects the learning strategies and we consider two common types of filters, defined in spectral space as

$$\text{Gaussian filter}: \quad G_\delta(k) = \exp\left(-\frac{k^2\Delta^2_{\text{reduced}}}{24}\right), \tag{25}$$

$$\text{Cut-off filter}: \quad G_\delta(k) = 0, \forall k > \pi\Delta^{-1}_{\text{reduced}}. \tag{26}$$

Regarding numerical aspects, we can solve the time integration of the reduced system g with a larger time-step by a factor corresponding to the grid size ratio (or filter scale δ), that is, $\Delta t_{\text{reduced}} = \delta\Delta t_{\text{true}}$. To generate the corresponding data sets $\{R(\mathbf{y}_i), \bar{\mathbf{y}}_i\}_n$ and $\{\mathbf{y}(t)\}_{t\in[0,T]}$, we subsample one true state every δ iterations performed by the true system f.

3.3. Baseline Parametrizations

For benchmarking purposes, we implement some physics-based baselines and focus on parametrizations based on functional eddy viscosity (Kraichnan, 1976), that is, models that artificially dissipate energy at relevant scales to remain stable. This is to be contrasted with structural models that produce backscatter and thus suffer from stability issues and will not be considered here. One can state these parameterizations in a flux formulation,

$$\mathcal{M}_{\text{P}}(\bar{\psi}, \bar{\omega}) = \nabla \cdot (\nu_e \nabla\bar{\omega}). \tag{27}$$

where the eddy viscosity coefficient ν_e contains an arbitrary constant c_{P} for which the optimal value depends on the flow configuration,

$$\nu_e = (c_{\text{P}}\Delta)^n |\varepsilon_{\text{P}}(\bar{\psi}, \bar{\omega})| \tag{28}$$

with n depending on the scaling law used to derive the model. We used the dynamic procedure proposed by Germano et al. (1991) and Lilly (1992) where the constant is computed from a least-square minimization of the residual SGS term with a filter size larger than δ, that is, $\tilde{\omega} = \omega * G_{\tilde{\delta}}, \tilde{\delta} > \delta$. We also apply spatial averaging with positive clipping (Guan, Chattopadhyay, et al., 2022; Pawar et al., 2020) in order to avoid locally negative constants $c_{\text{P}}(x, y) < 0$, that is, ensuring that the models are purely diffusive and $\nu_e \geq 0$.

One of the most popular SGS model has been proposed by Smagorinsky (1963). It derives from the assumption of direct cascade of energy, which is relevant for three-dimensional flows. However, this assumption is expected not to translate well to two-dimensional or geophysical turbulence, even if it has been already employed in global climate models (Delworth et al., 2012). Following a similar derivation, the Leith model (Leith, 1996) is often referred as the two-dimensional counterpart of the Smagorinsky model, assuming a direct cascade of enstrophy. The models are defined as eddy viscosity coefficients proportional to the resolved strain rate $\bar{S}$ and vorticity gradient $\nabla\bar{\omega}$, respectively,

$$\text{Smagorinsky model}: \quad \nu_e = (c_{\text{S}}\Delta)^2|\bar{S}|, \tag{29}$$

$$\text{Leith model}: \quad \nu_e = (c_{\text{L}}\Delta)^3|\nabla\bar{\omega}|. \tag{30}$$

We will denote by $\mathcal{M}_{\text{DynSmagorinsky}}$ and $\mathcal{M}_{\text{DynLeith}}$ the dynamic versions of these two models where c_{P} has been computed using the dynamic procedure mentioned above.

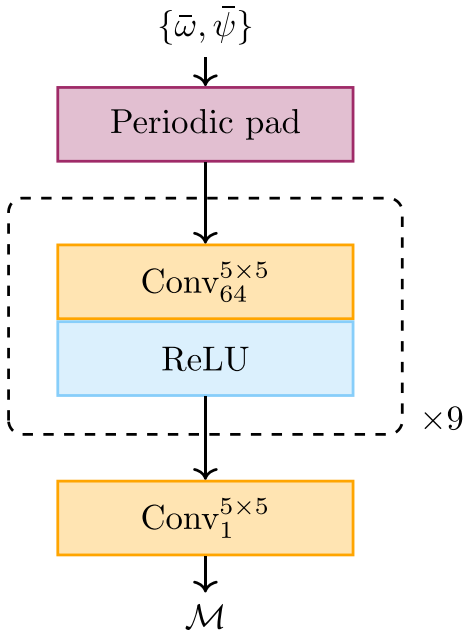

Figure 2. Sketch of the fully convolutional architecture employed in this study. The model applies nine inner convolutional blocks, that is, one 2D convolution layer Conv_C^K and one nonlinear ReLU activation. The number of channels C for the inner convolutions and the kernel size K are equal to 64 and 5, respectively. For the final layer, the model uses a 2D convolution layer to output the targeted real-valued subgrid-scale (SGS) term. Since our computational domain is doubly periodic, we also use a periodic padding as a first layer.

3.4. Neural Architecture and Training

Our main focus being here the impact of a priori and a posteriori strategies, we consider the same neural-network-based parametrization $\mathcal{M}$ with the two learning settings. We use a convolutional neural network architecture, which is particularly relevant for translation-invariant problems and have been used with success to train SGS parametrizations, for example, Beck et al. (2019), Bolton and Zanna (2019), Frezat et al. (2021), Guan, Chattopadhyay, et al. (2022), Lapeyre et al. (2019), and Mohan et al. (2020).

As shown in Figure 2, we use a simple ConvNet with 10 layers of convolutions with nonlinear ReLU activations. More involved architectures could further improve the performance. We may point out that our goal is not to design an optimal NN-based architecture, but rather to evaluate the impact of different learning strategies at similar computational complexity for the SGS parameterization.

Regarding the learning phase, the training loss for the a priori strategy (Equation 3) computes the MSE of the predicted term with respect to the true SGS term on a batch of S samples,

$$\mathcal{L}_{\text{prio}}(\mathcal{M}) := \frac{1}{S}\sum_{s=1}^{S} \| R(\psi,\omega)_s - \mathcal{M}(\bar{\psi}_s, \bar{\omega}_s) \|_2^2. \tag{31}$$

For the a posteriori strategy (Equation 7), the choice of training loss is more flexible, since we can explore spatio-temporal metrics for a batch of $N = T/\Delta t_{\text{reduced}}$ discrete integration steps. To illustrate the basics of the strategy, we choose the MSE of the most important state of the QG system, that is, the vorticity,

$$\mathcal{L}_{\text{post}}(\mathcal{M}) := \frac{1}{N}\sum_{n=1}^{N} \| \mathcal{T}(\omega(n\Delta t_{\text{reduced}})) - \bar{\omega}(n\Delta t_{\text{reduced}}) \|_2^2. \tag{32}$$

The models are trained with the Adam optimizer on the same data set containing 10 independent trajectories, that is, simulations of 3,000 snapshots each using different initial conditions, which gives a data set of 30,000 samples. As stated in Equation 32, we subsample training data from the DNS sequences with a $\Delta t_{\text{reduced}}$ sampling rate. We may point out that this configuration leads to a high inter-sample correlation for the training data set (Guan, Subel, et al., 2022). In our experiments, we chose an a priori batch size equal to the number of a posteriori temporal iterations, that is, $S = N = 25$. This experimental setting ensures that both a priori and a posteriori training epochs use each training state only once within one epoch. It enables a fair comparison of the two schemes in terms of training convergence. Empirically, we noted that 30 epochs were necessary for the a priori strategy, while five are enough for the a posteriori strategy. We note however that the latter is consequently more expensive due to the solver steps involved in the training loop. Regarding the a posteriori strategy, training a model that performs the time integration of a system of PDEs inside the minimization loop may lead to instabilities and difficulties. To address these issues, we consider Algorithm 1. It involves the following key steps:

- A gradual increase of the time horizon $[0, T]$ (Line 2). The time integration scheme of reduced system with operators g and $\mathcal{M}_{\text{NN}}$ may result in a very deep computational graph (typically, ConvNet with more than eight layers with the considered configuration with 25 integration steps), which may in turn lead to the commonly known vanishing gradient problem (Hochreiter et al., 2001), especially for the first epochs of the training process. To address this issue, we gradually increase the temporal horizon from $\left[0, \frac{T}{\epsilon}\right]$ to $[0, T]$ where ϵ corresponds to the number of training epochs. In this study, the increment is done using a simple linear function, but any increasing heuristic should work as long as the first critical epochs take a small number of iterations.
- Withdrawing simulated data on the fly for which the Courant-Friedrichs-Lewy (CFL) is greater than some threshold (Line 8). Incorrect predictions from the NN especially during the first epochs of the training process for PDE problems at the limit of numerical stability can lead to numerical blowups of the system and by

Algorithm 1. Training Algorithm for SGS Model $\mathcal{M}$ Using the a Posteriori Strategy. True Variables $\mathbf{y}$ are Sampled Randomly From Data Set Which is Only Required to Contain True Variables. In Practice, the Projection can Be Applied Beforehand and Data Set $\{\mathcal{T}(\mathbf{y}(t))\}_{t\in[0,T]}$ can Also Be Built From Projected True States. Note That the Outer Loop Iterates Over the Entire Trajectories for Each Epoch.

Require: dataset $\{\mathbf{y}(t)\}_{t\in[0,T]}$

Require: reduced system g, training model $\mathcal{M}(\bar{\mathbf{y}}|\theta)$

Require: number of iterations N, number of epochs ϵ, starting time t_0

Require: loss function $\mathcal{L}_{\text{post}}$

for $i \leftarrow 1$ to ϵ **do**
  $N_{\text{eff}} \leftarrow i \left\lfloor \frac{N}{\epsilon} \right\rfloor$ ▷ Define effective number of iterations
  $\{\mathbf{y}(t)\} \leftarrow \text{sample}(\{\mathbf{y}(t)\}, N_{\text{eff}})$ ▷ Randomly sample consecutive true states
  $\bar{\mathbf{y}}_0 \leftarrow \mathcal{T}(\mathbf{y}(t_0))$ ▷ Define initial state from true states
  for $j \leftarrow 1$ to N_{eff} **do**
    $\bar{\mathbf{y}}_j \leftarrow \mathcal{I}(g(\bar{\mathbf{y}}_{j-1}) + \mathcal{M}(\bar{\mathbf{y}}_{j-1}|\theta))$ ▷ Time integration with $\mathcal{I}$ (here RK4)
  end for
  if $\max_{j\in N_{\text{eff}}} \text{CFL}(\bar{\mathbf{y}}_j) > 1$ **then** ▷ Discard batch if stability is not obtained
    continue
  end if
  $\theta \leftarrow \text{step}(\mathcal{M}, \frac{\partial \mathcal{L}_{\text{post}}(\mathbf{y},\bar{\mathbf{y}})}{\partial \theta})$ ▷ Optimize model parameters from *a posteriori* loss
end for

consequence exploding gradient for the minimization algorithm. We then discard batches for which the CFL is greater than some threshold, commonly chosen to be 1.

4. Results

In order to evaluate the performance of a priori and a posteriori learning strategies, we report numerical experiments for three different configurations of QG flows (see Equation 3). First, we study decaying turbulence and compare the a posteriori strategy with previous works based on the a priori strategy (Guan, Chattopadhyay, et al., 2022; Maulik et al., 2019). Then, we assess the performance of the proposed models in a more realistic wind-forced configuration representative of mesoscale oceanic simulations (Fox-Kemper & Menemenlis, 2008; Graham & Ringler, 2013). Finally, we analyze the impact of planetary rotation through the beta-plane effect on a mid-latitude geophysical flow.

For these three cases, we consider the following experimental setup. The training and test data involve respectively 10 and 5 DNS corresponding to the same configurations with different initial conditions. The reduced systems are run with $\delta = 16$, that is, the reduced grid is 16 times smaller compared to the true grid resolution in each direction. Reduced systems are integrated for 6,000 iterations for the non-stationary decay cases and 18,000 iterations to determine long-term statistics of the forced and beta-plane configurations. We may emphasize that the simulations used for evaluation purposes are never seen during the training phase. The parameters of the different flows are shown in Table 1 in dimensionalized units. Overall, for each QG configuration, we report a

Table 1
Parameters of the Different Direct Numerical Simulations (DNS) Flow Configurations

Name	$N_x \times N_y$	$L_x \times L_y$	Δt	μ	ν	β	Re
Decay	$2{,}048 \times 2{,}048$	$2\pi \times 2\pi$	10^{-4}	0	3.125×10^{-5}	0	3.2×10^4
Forced	$2{,}048 \times 2{,}048$	$2\pi \times 2\pi$	10^{-4}	2×10^{-2}	1.025×10^{-5}	0	2.2×10^5
Beta-plane	$2{,}048 \times 2{,}048$	$2\pi \times 2\pi$	10^{-4}	2×10^{-2}	1.025×10^{-5}	2.195×10^2	3.4×10^5

Note. Details about simulation spin-up, initialization, and forcing parameters are described in more details in Section 4.2. Note that reduced systems use the same parameters, except for grid resolution $\left(\bar{N}_x, \bar{N}_y\right)$ obtained from the spatial filter scale $(N_x/\delta, N_y/\delta)$ and time-step $\Delta t_{\text{reduced}} = \delta \Delta t_{\text{true}}$. The quantities are given as numerical (unitless) values directly used in the solver for reproducibility. Still, details on how these parameters are chosen are provided in their corresponding result section.

quantitative synthesis for the cutoff and Gaussian filters and further illustrate the key features of the different learning strategies using a cutoff filter.

Important quantities discussed in the evaluation are both;

- **short-term** temporal evolution of quadratic invariants both for the energy (Equation 12) and the enstrophy (Equation 13), typical in weather forecast.
- **long-term** statistics from enstrophy spectrum and flux (Equation 17) in spectral space, relevant for climate predictions.

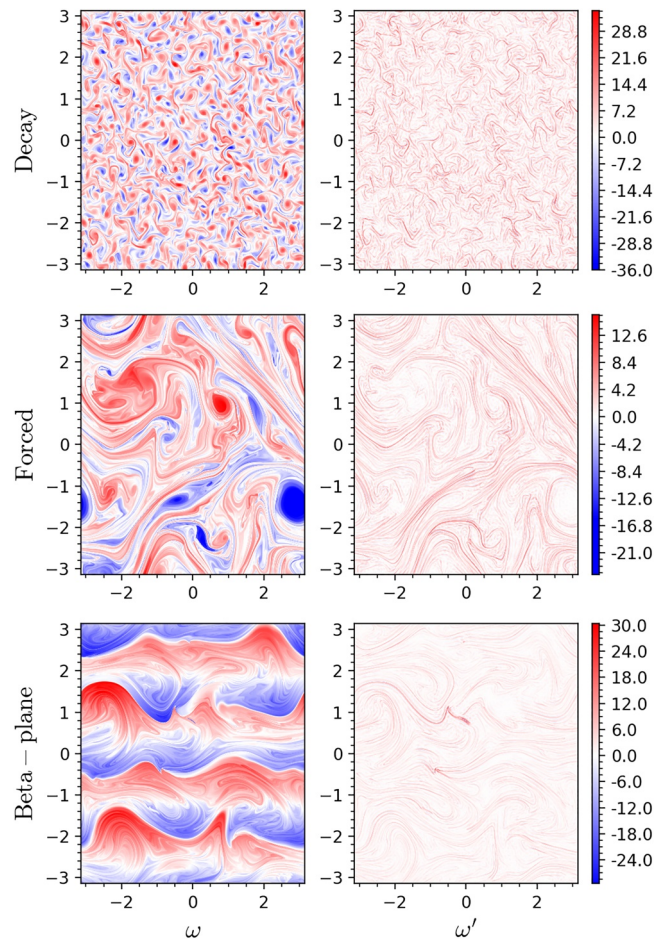

Figure 3. True initial vorticity field ω (left) and corresponding sub-grid contribution $\omega' = \omega - \bar{\omega}$ (right) for the three case-studies: decay (top), forced (middle) and beta-plane (bottom).

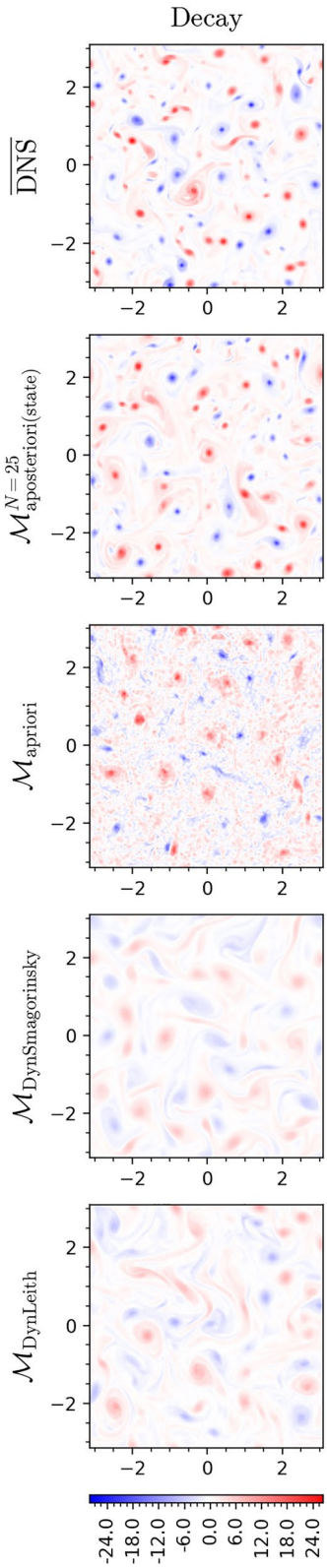

Figure 4. Vorticity fields for the different models at the end (6,000 reduced system iterations equivalent 96,000 true system iterations) of one decaying turbulence evaluation trajectory.

4.1. Decaying Turbulence

In the context of two-dimensional SGS parametrization with ML models, the decaying turbulence configuration is one of the most studied (Guan, Chattopadhyay, et al., 2022; Maulik et al., 2019; Pawar et al., 2020). This type of flow is particularly interesting because of its non-stationary nature, i.e., the system invariants are temporally varying. Similarly to Guan, Chattopadhyay, et al. (2022), we sample the initial vorticity fields randomly from a Gaussian distribution $\omega \sim \mathcal{N}(0, 1)$ at moderate wavenumbers $k \in [10, 32]$ and integrate the system for 10,000 iterations before reaching spectrum self-similarity (Batchelor, 1969).

From this starting time t_0, the vorticity fields exhibit an early turbulence behavior with a lot of fine structures (see Figure 3, top). DNS and reduced models with SGS parametrizations are run until $t = 9.6$, which is longer than the temporal horizon used in the training data by a factor of two. In two-dimensional decaying flows, we expect to see vortex pairing and emergence of larger structures as shown in Figure 4. Due to their purely diffusive form, parameterizations $\mathcal{M}_{\text{DynSmagorinsky}}$ and $\mathcal{M}_{\text{DynLeith}}$ cannot perform well on this configuration and are indeed incorrectly dissipating relevant small scales. The NN-based model trained with the a priori strategy has accumulated small-scale enstrophy and is thus perturbed with noise coming from numerical instabilities. The model trained with the a posteriori visually shows the expected stable dynamics with small-scale features, even outside the training regime, which supports some degree of generalization (or extrapolation).

The evolution of domain-averaged quadratic integrals in Figure 5 confirms the observations from the vorticity fields, since we can see a large energy and enstrophy decrease for both $\mathcal{M}_{\text{DynSmagorinsky}}$ and $\mathcal{M}_{\text{DynLeith}}$, while the a priori model correctly captures the energy decay but dissipates enstrophy too slowly compared to the DNS. Spectral statistics shown in Figure 6 are also in close agreement to the DNS for the a posteriori-trained model, in particular for the large wavenumbers of the enstrophy spectrum $Z(k) = k^2 E(k)$ which particularly highlight the dynamics of the smallest resolved scales.

4.2. Forced Turbulence

The second case-study involves QG flows with a source term F designed to mimic wind-stress. We study a particular configuration inspired by Graham and Ringler (2013), which evaluated the performance of a large number of physics-based parametrizations in mesoscale ocean simulations. To reproduce these realistic equilibrium solutions, we use a bottom drag ($\mu > 0$) and initiate turbulent mixing from a wind-stress slowly varying in time at large-scale $k = 4$ with steady enstrophy rate injection $Z(F) = 3$ such that,

$$F_\omega(t) = \cos(4y + \pi \sin(1.4t)) - \cos(4x + \pi \sin(1.5t)) \tag{33}$$

$$F = \frac{1}{3} Z(F_\omega) F_\omega(t) \tag{34}$$

In order to converge to a stationary turbulent state, we initialize the simulation runs from a few large-scale Fourier modes and spin-up on a smaller grid ($1{,}024^2$) for over 500,000 iterations. The initial conditions for training and

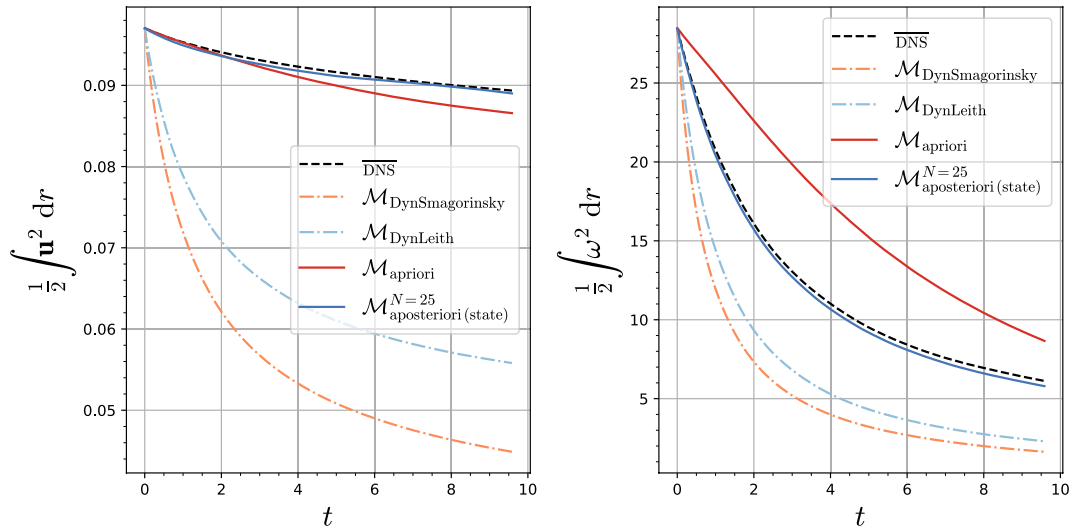

Figure 5. Evolution of domain-averaged energy (left) and enstrophy (right) computed in non-dimensionalized time units in the decaying turbulence setting.

evaluation (see Figure 3, middle) are taken after energy and enstrophy propagation to the smallest scales of the true grid ($2{,}048^2$) in about 25,000 iterations.

To evaluate the long-term performance of the forced configuration in equilibrium, we run simulations until $t = 2.88$, which is at least three times longer than the complete decorrelation time of the system due to chaos. We report the vorticity fields at the end of the simulations in Figure 7. The true vorticity state exhibits both large vortices generated by the wind forcing and small filaments in between. Overall, we draw conclusions similar to the decaying turbulence regime. We note that the small structures are inaccurately predicted for both $\mathcal{M}_{\mathrm{DynSmagorinsky}}$ and $\mathcal{M}_{\mathrm{DynLeith}}$ due to dissipation and for the a priori model due to numerical instabilities. By contrast, the a posteriori model is the only one to correctly capture both the large-scale and fine-scale patterns in this configuration.

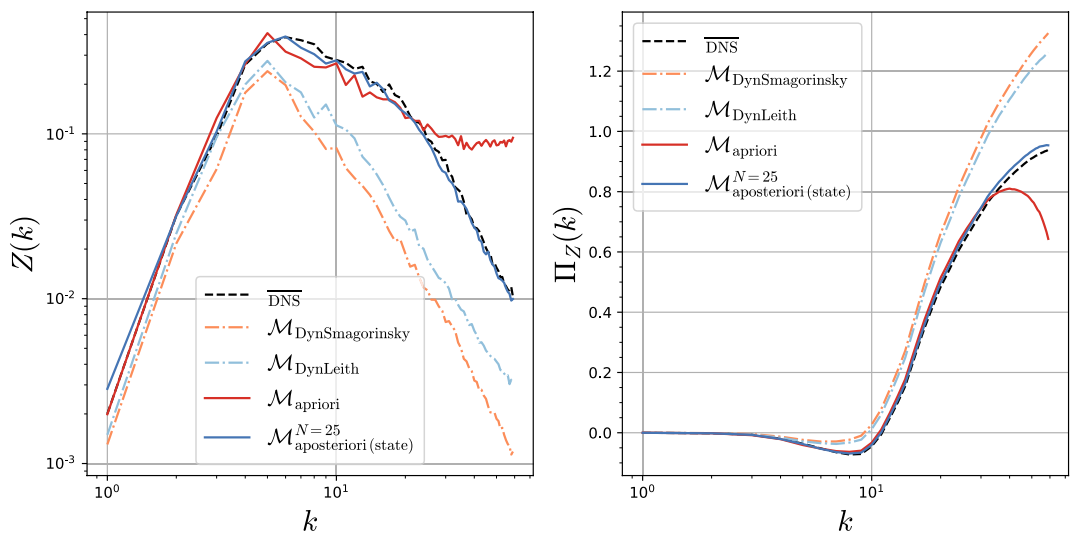

Figure 6. Final enstrophy spectrum (left) and time-averaged enstrophy flux (right) describing statistical performance of the models in decaying turbulence.

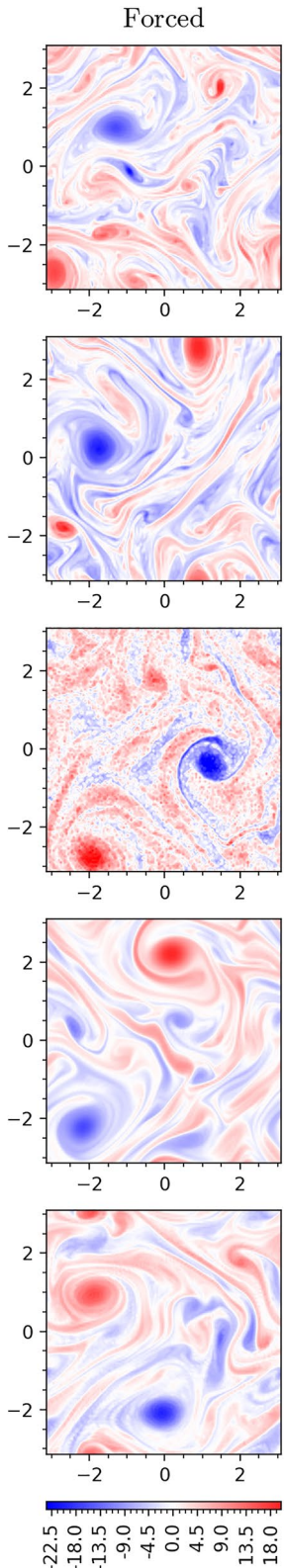

Figure 7. Vorticity fields for the different models at the end (18,000 reduced system iterations equivalent 288,000 true system iterations) of one forced turbulence evaluation trajectory.

While domain-averaged integrals fluctuate a lot on such long-term trajectories due to the chaotic nature of the flow, we expect those quantities to remain approximately constant over time. This property is verified on the kinetic energy for both NN-based models, but the enstrophy of the a priori-trained model increases over time, which indicates some accumulation of small-scale energy (see Figure 8) and may result in a potential future blow-up of the simulation. However, the time-averaged statistical enstrophy spectrum shown in Figure 9 demonstrates the ability of the a posteriori model to reproduce accurately both the smallest scales and the largest scales of the simulation (small wavenumbers) compared to the other models.

4.3. Beta-Plane Turbulence

The third case-study runs the same forced simulation as in the previous configuration complemented by a beta-plane effect to account for the meridional variation of Coriolis force caused by a spherical shape. We take a Rossby parameter β corresponding to Earth's planetary rotation on mid-latitudes (60°).

The beta-effect has an important impact on the topology of the dynamics, as it creates high-velocity longitudinal jets as seen in Figure 3, bottom. In this simple setting without topography (flat bottom layer), the system does not go through state transitions but remains in a statistical equilibrium. The strong vorticity gradients in between jets are predicted more accurately (see Figure 10) by the $\mathcal{M}_{\text{DynLeith}}$ than the $\mathcal{M}_{\text{DynSmagorinsky}}$, which still over-dissipates at small-scale. The model trained a priori is not stable at all in this configuration, while the a posteriori model remains stable to simulate visually consistent patterns.

The instabilities visible in the vorticity field of the a priori model are explained by the increasing energy and enstrophy in Figure 11. This reveals a non-conservative behavior which leads to a simulation blowup. The a posteriori strategy performs extremely well on this long-term simulation, predicting in particular a correct enstrophy evolution compared to that of the DNS. The enstrophy spectrum and fluxes (see Figure 12) are similar to those of the forced turbulence configuration, except that the linear damping has a stronger impact, due to the relative increase in velocity from the beta-effect. Overall, the conclusions are the same as the previous two configurations, with highest fidelity small-scale dynamics being produced by the model trained using the a posteriori strategy.

4.4. Quantitative Analysis

As a quantitative synthesis of our numerical experiments, we first report the performance on two metrics for the three case-studies and the different models: an a priori metric given by Pearson correlation coefficient of the predicted SGS terms (Table. 2.) and an a posteriori metric given by a variant of the error-landscape enstrophy flux assessment presented by Meyers (2011) (Table. 3). We report these performance metrics both for cutoff and Gaussian filters.

The selected learning strategy clearly impacts the corresponding metrics. For the three flow regimes, the a priori learning performs better on a priori metrics, whereas the a posteriori learning leads to the best score

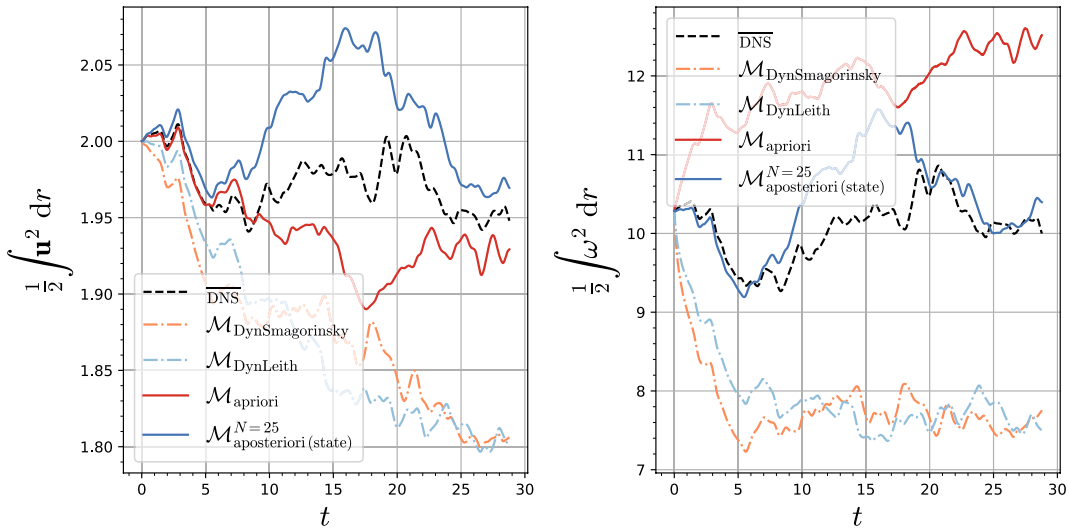

Figure 8. Evolution of domain-averaged energy (left) and enstrophy (right) computed in non-dimensionalized time units in the forced turbulence setting.

on a posteriori metrics. Concerning the projection kernel, physical models and the NN-based model trained a priori perform better with the Gaussian filter. Note that this is not surprising, as Gaussian filtering tends to smooth out discontinuities in the subgrid term (Canuto et al., 2007) which might help during a priori training. By contrast, the a posteriori learning scheme results in consistent scores for both the Gaussian and cutoff kernels.

We analyze the ability of the benchmarked models to reproduce modeled transfers in physical space. Figure 13 shows the PDF of $T_{\bar{z}}$ for the three different configurations at the end of their respective trajectories. First, while we can not explicitly identify forward and backward transfers, we can see that the resolved transfer term produced by the physical models is not symmetric, that is, it produces a larger amount of negative values. This suggests that theses models are not able to produce backscatter by construction. The a posteriori model always

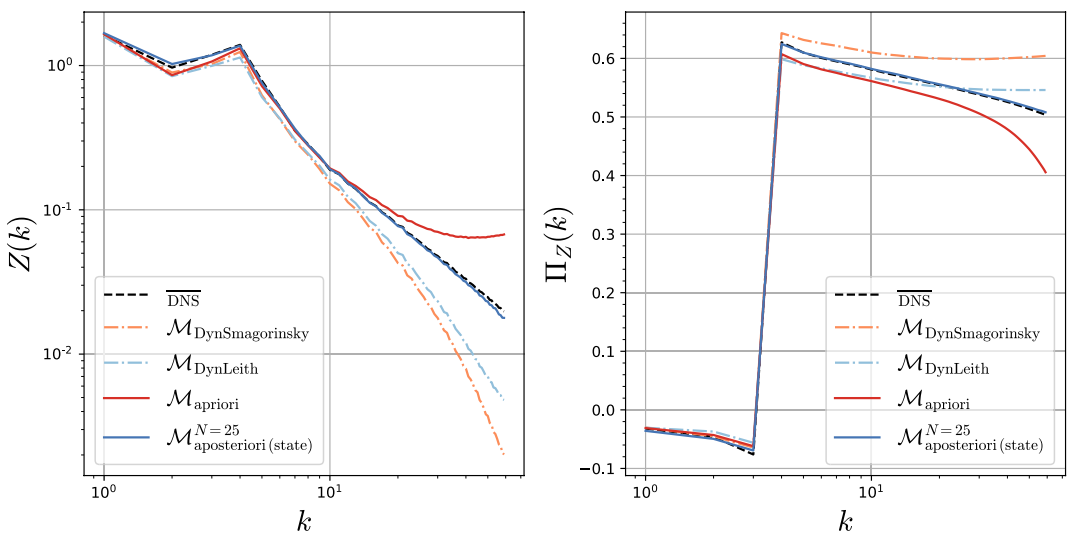

Figure 9. Time-averaged enstrophy spectrum (left) and enstrophy flux (right) describing statistical performance of the models in forced turbulence.

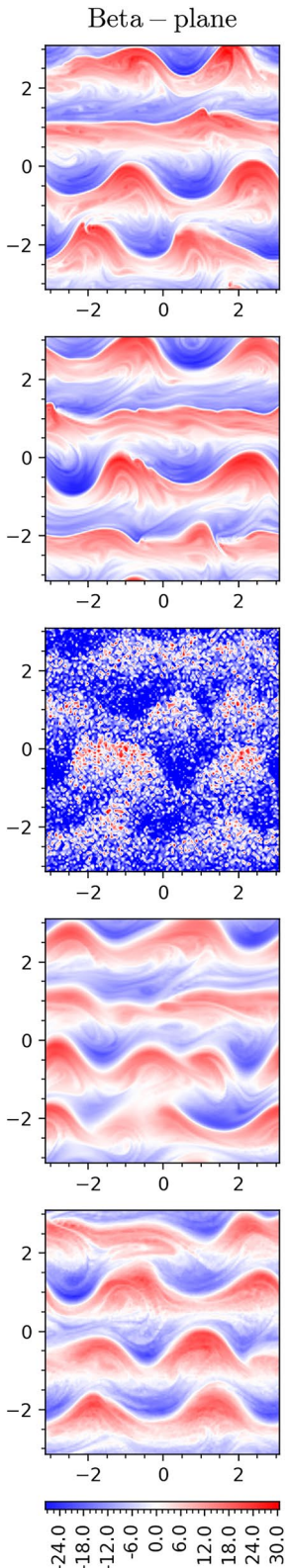

Figure 10. Vorticity fields for the different models at the end (18,000 reduced system iterations equivalent 288,000 true system iterations) of one beta-plane turbulence evaluation trajectory.

performs better than the a priori one, with predictions that better reproduce the DNS on the tails of the distribution. However, even if the a priori model has been demonstrated to be unstable on the beta-plane configuration, the resolved transfer term remains close to the DNS, but overpredicts positive tails, which, again, could indicate an incorrect representation of energy backscatter. We can further analyze the instability related to $T_{\bar{z}}$ by looking at the evolution of its spatial average, as shown in Figure 14. The resolved transfer term of the a priori model rapidly diverges from the DNS in the beta-plane configuration, which identifies the instability. As expected, the physical models produce large negative values at the beginning of the simulations, which can be explained by their highly dissipative nature. The a posteriori model, however, is the most accurate and remains close to the DNS throughout the trajectories.

5. Discussion

This study investigated different learning strategies to train SGS parametrizations for two-dimensional QG turbulent flows. While the state-of-the-art has mostly explored a priori learning schemes, our numerical experiments stress the significant improvement brought by the a posteriori learning strategy to better reproduce small-scale dynamics on large temporal horizons with great accuracy. For all the flow configurations and coarsening schemes considered in this study, SGS parametrizations trained according to a posteriori training loss clearly outperform both physics-based and ML baselines.

The a posteriori learning strategy introduced in this paper opens the possibility to design stable subgrid parametrizations with more flexibility than state-of-the-art ML-based approaches. Indeed, we have here explored a relatively simple a posteriori training loss given by a vorticity-based MSE, the a posteriori learning scheme offers a much greater flexibility for the exploitation and combination of different a posteriori metrics during the learning phase. Losses defined from classic performance metrics such as energy transfers and distributions seem particularly appealing. One may also explore application-specific metrics including among others boundary layers flows. As the a posteriori learning strategy results in an improved stability of the trained SGS parametrizations, it may also offer means to explore more complex neural architectures for SGS terms. Here, we considered a relatively simple ConvNet, but more complex and state-of-the-art neural architectures including for instance ResNet, UNet and transformer networks could be worth exploring. Another interesting avenue is the joint training of a posteriori models in the context of data assimilation such as described in Bonavita and Laloyaux (2020) and Farchi et al. (2021).

An interesting connection can indeed be made between a posteriori learning and variational data assimilation techniques. Our a posteriori learning algorithm formulates a variational problem which is formally equivalent to the strong constraint 4D-Var scheme (Blayo et al., 2015; Carrassi et al., 2018). But, in our case, the *control vector* is composed of the parameters of the neural network and observations are assumed to be perfect. The analogy between a posteriori learning and 4D-Var therefore brings the question of whether parametrizations, or more generally corrections to existing models, could be learned directly from sparse and noisy observa-

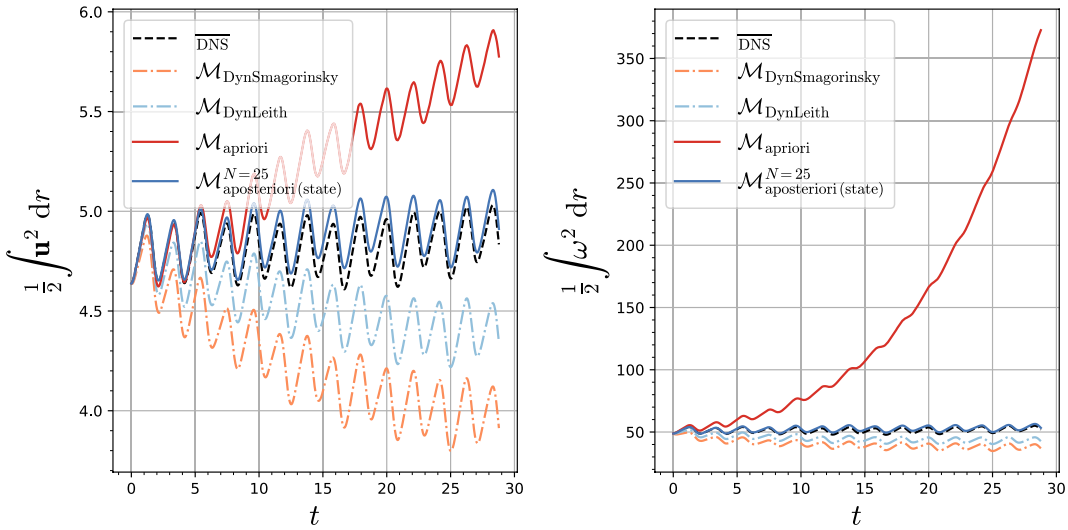

Figure 11. Evolution of domain-averaged energy (left) and enstrophy (right) computed in non-dimensionalized time units in the beta-plane turbulence setting.

tions (Schneider, Lan, et al., 2017). In this sense, a posteriori learning is related to the bias correction methods that have been proposed in data assimilation (Dee, 2005), and especially the schemes proposed to infer state-dependent corrections to existing models (D'andrea & Vautard, 2000; Griffith & Nichols, 2000). Interestingly, this field has received renewed attention over recent years with several authors proposing to approach bias correction with ML (Bonavita & Laloyaux, 2020; Farchi et al., 2021). In this context, we stress that our approach is very similar to the scheme introduced by Farchi et al. (2021), with the noticeable difference that we here learn a correction through the 4D-Var scheme itself, and not from the increment of the assimilation scheme.

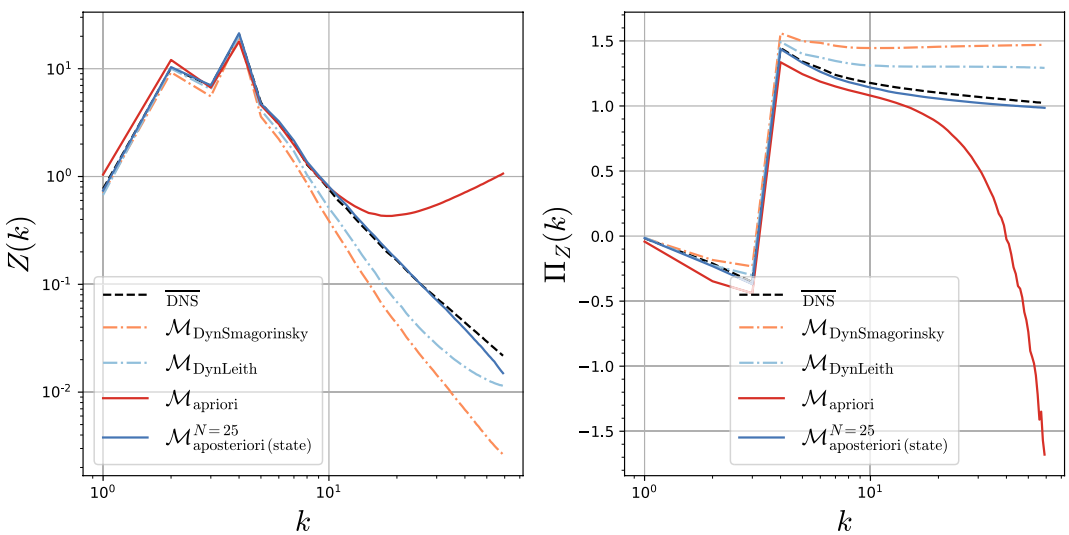

Figure 12. Time-averaged enstrophy spectrum (left) and enstrophy flux (right) describing statistical performance of the models in beta-plane turbulence.

Table 2
Short-Term Performance of the Considered Subgrid-Scale (SGS) Parametrizations in the Three Different Configurations With Both Cutoff and Gaussian Projection Kernels

	Decay		Forced		Beta-plane	
$\rho_{R,\mathcal{M}}$	Cutoff	Gaussian	Cutoff	Gaussian	Cutoff	Gaussian
$\mathcal{M}_{\text{DynSmagorinsky}}$	0.16	0.38	0.09	0.55	0.04	0.28
$\mathcal{M}_{\text{DynLeith}}$	0.13	0.32	0.08	0.49	0.03	0.17
$\mathcal{M}_{\text{a priori}}$	0.75	**0.90**	**0.82**	**0.95**	**0.82**	**0.96**
$\mathcal{M}_{\text{a posteriori (states)}}$	**0.77**	0.57	0.45	0.29	0.48	0.21

Note. We compute the correlation coefficient between predicted and exact subgrid terms, which favors the a priori learning strategy.

By construction, parametrizations learned with the a posteriori strategy should improve the short-term forecasting capabilities of the models. This is true in particular if the training loss is defined as the sum of forecasting errors over a given time horizon as considered here. Interestingly, we noted here that for SGS parametrizations, this short-term forecasting performance translates in a better long-term stability and representation of long-term flow patterns where the long-term horizon being several order of magnitudes greater than the time horizon in the training loss (18,000 vs. 25 time steps). While recent studies have explored neural models for the short-term forecasting of realistic geophysical flows, especially for weather forecasting applications (Schultz et al., 2021; Weyn et al., 2021), we believe our study opens new avenues for the exploitation of learning-based components in climate-scale simulations, which remain an open challenge (Rasp et al., 2018). In this respect, to account for the chaotic nature of turbulent flows, a posteriori training losses could also benefit from statistical metrics as opposed to synoptic ones as the MSE used in this work.

But a strong requirement of the proposed framework lies in the differentiability of the considered dynamical model, which may question its practical applicability. Indeed, most large-scale forward solvers in earth system models (ESM) rely on high-performance languages that do not embed AD capabilities. While it is generally recognized that adjoint models are very useful additional tools for these solvers (Barkmeijer, 2009; Wunsch & Heimbach, 2013), adjoint operators are readily available only for a small fraction of them (Heimbach et al., 2005; Vidard et al., 2015). We stress that the emergence of a new generation of models written in differentiable programming languages such as JAX and JuliaDiff (Häfner et al., 2021; Huang &

Table 3
Long-Term Performance of Considered Subgrid-Scale (SGS) Parametrizations in the Three Different Configurations With Both Cutoff and Gaussian Projection Kernels

	Decay		Forced		Beta-plane	
$L^2\left(\Pi_Z^R - \Pi_Z^{\mathcal{M}}\right)$	Cutoff	Gaussian	Cutoff	Gaussian	Cutoff	Gaussian
$\mathcal{M}_{\text{DynSmagorinsky}}$	1.95	1.31	0.49	0.16	2.83	1.75
$\mathcal{M}_{\text{DynLeith}}$	1.64	1.02	0.16	0.11	1.66	0.98
$\mathcal{M}_{\text{a priori}}$	0.74	0.60	0.36	0.40	8.73	0.26
$\mathcal{M}_{\text{a posteriori (states)}}$	**0.13**	**0.09**	**0.02**	**0.02**	**0.30**	**0.05**

Note. We compute the L^2 distance between the reference enstrophy fluxes and the ones simulated using the different SGS parameterizations. The a posteriori learning strategy clearly leads to much better scores.

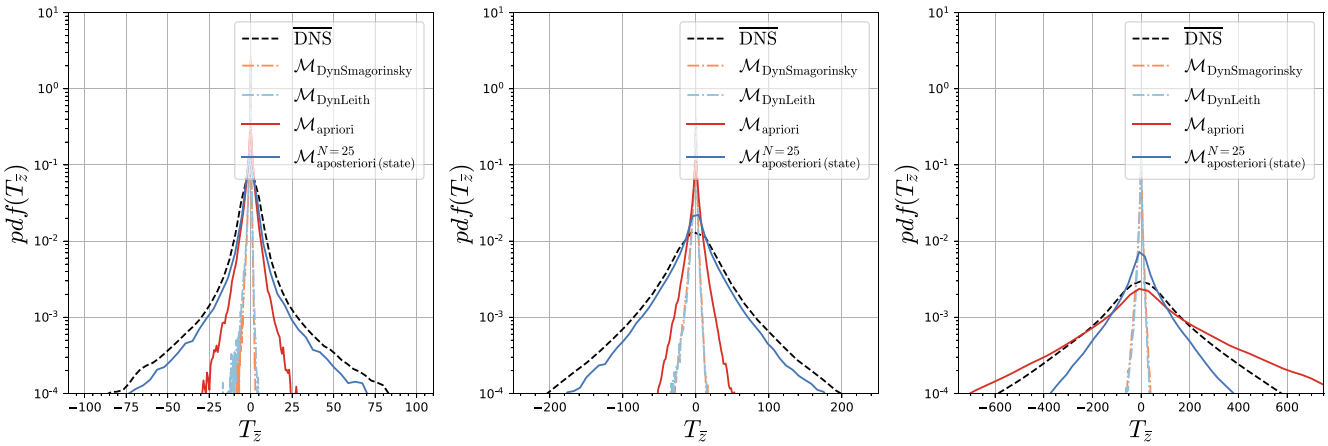

Figure 13. Probability distribution of the modeled transfer term in physical space for the three case-studies: decay (left), forced (middle), and beta-plane (right) at the end of the trajectories.

Topping, 2021; Ramadhan et al., 2020; Sridhar et al., 2021) naturally supports our contribution. Besides, deep differentiable emulators (Hatfield et al., 2021; Kasim et al., 2021; Nonnenmacher & Greenberg, 2021) that learn a differentiable approximation of a non-differentiable forward solver or of its adjoint may also open new avenues for the development of SGS parametrizations for state-of-the-art ESMs with a posteriori learning strategies.

Still, while one could benefit from the ongoing effort for differentiable ESMs, the development and the practical implementation of ML-based parametrizations in non-differentiable ESMs (Madec et al., 2017; Marshall et al., 1997) is also an important question. In this context, physics-based neural networks using a priori training schemes arise as relevant options, given the promising results already reported for the oceanic (Zanna & Bolton, 2021) and atmospheric (Gentine et al., 2021) parts of climate models. We also believe that the proposed a posteriori strategy combined with physics-based constraints will also be worth exploring in future work.

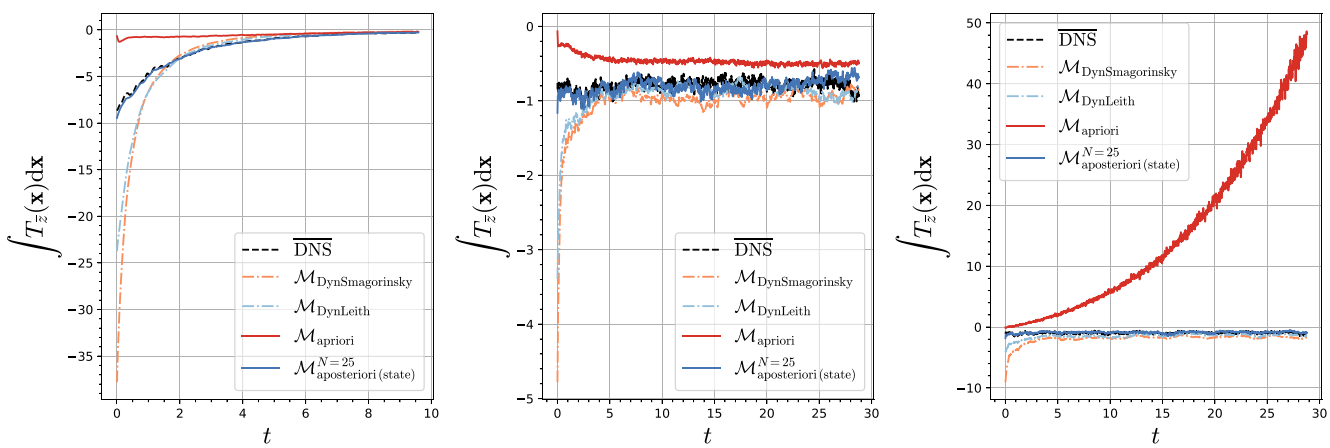

Figure 14. Temporal evolution of the spatially averaged modeled transfer term in physical space for the three case-studies: decay (left), forced (middle), and beta-plane (right).

Data Availability Statement

The results can be reproduced from the data, models, and associated learning algorithms provided along with the pseudo-spectral quasi-geostrophic code, available in https://doi.org/10.5281/zenodo.6799035.

Acknowledgments
The authors would like to thank Laure Zanna, Olivier Pannekoucke, Corentin Lapeyre and Emmanuel Cosme for helpful discussions. This research was supported by the CNRS through the 80 PRIME project and the ANR through the Melody, OceaniX, and HRMES ANR-17-MPGA-0010 projects. Additional support was also provided by Schmidt Futures, a philanthropic initiative founded by Eric and Wendy Schmidt, as part of its Virtual Earth System Research Institute (VESRI). Computations were performed using GPU resources from GENCI-IDRIS. The authors also wish to thank the reviewers for their insightful suggestions that helped to improve this manuscript.

References

Alet, F., Doblar, D., Zhou, A., Tenenbaum, J., Kawaguchi, K., & Finn, C. (2021). Noether networks: Meta-learning useful conserved quantities. *Advances in Neural Information Processing Systems*, *34*.

Bakarji, J., & Tartakovsky, D. M. (2021). Data-driven discovery of coarse-grained equations. *Journal of Computational Physics*, *434*, 110219. https://doi.org/10.1016/j.jcp.2021.110219

Barkmeijer, J. (2009). Adjoint diagnostics for the atmosphere and ocean. Retrieved from https://www.ecmwf.int/node/15243

Batchelor, G. K. (1969). Computation of the energy spectrum in homogeneous two-dimensional turbulence. *Physics of Fluids*, *12*(12), II–233. https://doi.org/10.1063/1.1692443

Bauer, P., Thorpe, A., & Brunet, G. (2015). The quiet revolution of numerical weather prediction. *Nature*, *525*(7567), 47–55. https://doi.org/10.1038/nature14956

Beck, A., Flad, D., & Munz, C.-D. (2019). Deep neural networks for data-driven LES closure models. *Journal of Computational Physics*, *398*, 108910. https://doi.org/10.1016/j.jcp.2019.108910

Blayo, É., Bocquet, M., Cosme, E., & Cugliandolo, L. F. (2015). *Advanced data assimilation for geosciences: Lecture notes of the LES Houches School of Physics: Special issue, June 2012*. Oxford: OUP.

Boffetta, G., & Ecke, R. E. (2012). Two-dimensional turbulence. *Annual Review of Fluid Mechanics*, *44*, 427–451. https://doi.org/10.1146/annurev-fluid-120710-101240

Bolton, T., & Zanna, L. (2019). Applications of deep learning to ocean data inference and subgrid parameterization. *Journal of Advances in Modeling Earth Systems*, *11*(1), 376–399. https://doi.org/10.1029/2018ms001472

Bonavita, M., & Laloyaux, P. (2020). Machine learning for model error inference and correction. *Journal of Advances in Modeling Earth Systems*, *12*(12), e2020MS002232. https://doi.org/10.1029/2020ms002232

Bouchet, F., & Venaille, A. (2012). Statistical mechanics of two-dimensional and geophysical flows. *Physics Reports*, *515*(5), 227–295. https://doi.org/10.1016/j.physrep.2012.02.001

Brenowitz, N. D., & Bretherton, C. S. (2018). Prognostic validation of a neural network unified physics parameterization. *Geophysical Research Letters*, *45*(12), 6289–6298. https://doi.org/10.1029/2018gl078510

Brunton, S. L., Noack, B. R., & Koumoutsakos, P. (2020). Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, *52*, 477–508. https://doi.org/10.1146/annurev-fluid-010719-060214

Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, *113*(15), 3932–3937. https://doi.org/10.1073/pnas.1517384113

Canuto, C., Hussaini, M. Y., Quarteroni, A., & Thomas, A., Jr. (2012). *Spectral methods in fluid dynamics*. Springer Science & Business Media.

Canuto, C., Hussaini, M. Y., Quarteroni, A., & Zang, T. A. (2007). *Spectral methods: Fundamentals in single domains*. Springer Science & Business Media.

Carati, D., Ghosal, S., & Moin, P. (1995). On the representation of backscatter in dynamic localization models. *Physics of Fluids*, *7*(3), 606–616. https://doi.org/10.1063/1.868585

Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., et al. (2019). Machine learning and the physical sciences. *Reviews of Modern Physics*, *91*(4), 045002. https://doi.org/10.1103/revmodphys.91.045002

Carrassi, A., Bocquet, M., Bertino, L., & Evensen, G. (2018). Data assimilation in the geosciences: An overview of methods, issues, and perspectives. *Wiley Interdisciplinary Reviews: Climate Change*, *9*(5), e535. https://doi.org/10.1002/wcc.535

Chantry, M., Hatfield, S., Dueben, P., Polichtchouk, I., & Palmer, T. (2021). Machine learning emulation of gravity wave drag in numerical weather forecasting. *Journal of Advances in Modeling Earth Systems*, *13*(7), e2021MS002477. https://doi.org/10.1029/2021ms002477

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. (2018). Neural ordinary differential equations. In *Advances in neural information processing systems* (Vol. *31*).

Cohen, T., & Welling, M. (2016). Group equivariant convolutional networks. In *International Conference on Machine Learning* (pp. 2990–2999).

Couvreux, F., Hourdin, F., Williamson, D., Roehrig, R., Volodina, V., Villefranque, N., et al. (2021). Process-based climate model development harnessing machine learning: I. A calibration tool for parameterization improvement. *Journal of Advances in Modeling Earth Systems*, *13*(3), e2020MS002217. https://doi.org/10.1029/2020ms002217

Cranmer, M., Greydanus, S., Hoyer, S., Battaglia, P., Spergel, D., & Ho, S. (2020). Lagrangian neural networks. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*.

Cushman-Roisin, B., & Beckers, J.-M. (2011). *Introduction to geophysical fluid dynamics: Physical and numerical aspects*. Academic Press.

D'andrea, F., & Vautard, R. (2000). Reducing systematic errors by empirically correcting model errors. *Tellus*, *52*(1), 21–41.

Danilov, S., Juricke, S., Kutsenko, A., & Oliver, M. (2019). Toward consistent subgrid momentum closures in ocean models. In *Energy transfers in atmosphere and ocean* (pp. 145–192). Springer. https://doi.org/10.1007/978-3-030-05704-6_5

de Avila Belbute-Peres, F., Smith, K., Allen, K., Tenenbaum, J., & Kolter, J. Z. (2018). End-to-end differentiable physics for learning and control. *Advances in Neural Information Processing Systems*, *31*, 7178–7189.

Dee, D. P. (2005). Bias and data assimilation. *Quarterly Journal of the Royal Meteorological Society*, *131*(613), 3323–3343. https://doi.org/10.1256/qj.05.137

Delworth, T. L., Rosati, A., Anderson, W., Adcroft, A. J., Balaji, V., Benson, R., et al. (2012). Simulated climate and climate change in the GFDL CM2.5 high-resolution coupled climate model. *Journal of Climate*, *25*(8), 2755–2781. https://doi.org/10.1175/jcli-d-11-00316.1

Fablet, R., Chapron, B., Drumetz, L., Mémin, E., Pannekoucke, O., & Rousseau, F. (2021). Learning variational data assimilation models and solvers. *Journal of Advances in Modeling Earth Systems*, *13*(10), e2021MS002572. https://doi.org/10.1029/2021ms002572

Farchi, A., Laloyaux, P., Bonavita, M., & Bocquet, M. (2021). Using machine learning to correct model error in data assimilation and forecast applications. *Quarterly Journal of the Royal Meteorological Society*, *147*(739), 3067–3084. https://doi.org/10.1002/qj.4116

Fox-Kemper, B., Adcroft, A., Böning, C. W., Chassignet, E. P., Curchitser, E., Danabasoglu, G., et al. (2019). Challenges and prospects in ocean circulation models. *Frontiers in Marine Science*, *6*, 65. https://doi.org/10.3389/fmars.2019.00065

Fox-Kemper, B., Bachman, S., Pearson, B., & Reckinger, S. (2014). Principles and advances in subgrid modeling for eddy-rich simulations. *Clivar Exchanges*, *19*(2), 42–46.

Fox-Kemper, B., & Menemenlis, D. (2008). Can large eddy simulation techniques improve mesoscale rich ocean models? *Ocean Modeling in an Eddying Regime*, *177*, 319–337. https://doi.org/10.1029/177gm19

Frezat, H., Balarac, G., Le Sommer, J., Fablet, R., & Lguensat, R. (2021). Physical invariance in neural networks for subgrid-scale scalar flux modeling. *Physical Review Fluids*, *6*(2), 024607. https://doi.org/10.1103/physrevfluids.6.024607

Gamahara, M., & Hattori, Y. (2017). Searching for turbulence models by artificial neural network. *Physical Review Fluids*, *2*(5), 054604. https://doi.org/10.1103/physrevfluids.2.054604

Gentine, P., Eyring, V., & Beucler, T. (2021). Deep learning for the parametrization of subgrid processes in climate models. *Deep Learning for the Earth Sciences: A Comprehensive Approach to Remote Sensing, Climate Science, and Geosciences*, 307–314. https://doi.org/10.1002/9781119646181.ch21

Germano, M., Piomelli, U., Moin, P., & Cabot, W. H. (1991). A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids*, *3*(7), 1760–1765. https://doi.org/10.1063/1.857955

Glasmachers, T. (2017). Limits of end-to-end learning. In *Asian Conference on Machine Learning* (pp. 17–32).

Graham, J. P., & Ringler, T. (2013). A framework for the evaluation of turbulence closures used in mesoscale ocean large-eddy simulations. *Ocean Modeling*, *65*, 25–39.

Griffith, A. K., & Nichols, N. K. (2000). Adjoint methods in data assimilation for estimating model error. *Flow, Turbulence, and Combustion*, *65*(3), 469–488. https://doi.org/10.1023/a:1011454109203

Guan, Y., Chattopadhyay, A., Subel, A., & Hassanzadeh, P. (2022). Stable a posteriori LES of 2D turbulence using convolutional neural networks: Backscattering analysis and generalization to higher Re via transfer learning. *Journal of Computational Physics*, *458*, 111090. https://doi.org/10.1016/j.jcp.2022.111090

Guan, Y., Subel, A., Chattopadhyay, A., & Hassanzadeh, P. (2022). Learning physics-constrained subgrid-scale closures in the small-data regime for stable and accurate LES. arXiv preprint arXiv:2201.07347.

Guillaumin, A. P., & Zanna, L. (2021). Stochastic-deep learning parameterization of ocean momentum forcing. *Journal of Advances in Modeling Earth Systems*, *13*(9), e2021MS002534. https://doi.org/10.1029/2021ms002534

Gupta, A., Jayaram, R., Chaterjee, A. G., Sadhukhan, S., Samtaney, R., & Verma, M. K. (2019). Energy and enstrophy spectra and fluxes for the inertial-dissipation range of two-dimensional turbulence. *Physical Review E*, *100*(5), 053101. https://doi.org/10.1103/physreve.100.053101

Häfner, D., Nuterman, R., & Jochum, M. (2021). Fast, cheap, and turbulent—Global ocean modeling with GPU acceleration in python. *Journal of Advances in Modeling Earth Systems*, *13*(12), e2021MS002717. https://doi.org/10.1029/2021ms002717

Hatfield, S., Chantry, M., Dueben, P., Lopez, P., Geer, A., & Palmer, T. (2021). Building tangent-linear and adjoint models for data assimilation with neural networks. *Journal of Advances in Modeling Earth Systems*, *13*(9), e2021MS002521. https://doi.org/10.1029/2021ms002521

Heimbach, P., Hill, C., & Giering, R. (2005). An efficient exact adjoint of the parallel MIT general circulation model, generated via automatic differentiation. *Future Generation Computer Systems*, *21*(8), 1356–1371. https://doi.org/10.1016/j.future.2004.11.010

Hochreiter, S., Bengio, Y., Frasconi, P., & Schmidhuber, J. (2001). *Gradient flow in recurrent nets: The difficulty of learning long-term dependencies*. IEEE Press.

Holl, P., Thuerey, N., & Koltun, V. (2020). Learning to control PDES with differentiable physics. In *International Conference on Learning Representations*.

Huang, L., & Topping, D. (2021). JlBox v1. 1: A julia-based multi-phase atmospheric chemistry box model. *Geoscientific Model Development*, *14*(4), 2187–2203. https://doi.org/10.5194/gmd-14-2187-2021

Jansen, M. F., Held, I. M., Adcroft, A., & Hallberg, R. (2015). Energy budget-based backscatter in an eddy permitting primitive equation model. *Ocean Modeling*, *94*, 15–26. https://doi.org/10.1016/j.ocemod.2015.07.015

Juricke, S., Danilov, S., Koldunov, N., Oliver, M., & Sidorenko, D. (2020). Ocean kinetic energy backscatter parametrization on unstructured grids: Impact on global eddy-permitting simulations. *Journal of Advances in Modeling Earth Systems*, *12*(1), e2019MS001855. https://doi.org/10.1029/2019ms001855

Kasim, M., Watson-Parris, D., Deaconu, L., Oliver, S., Hatfield, P., Froula, D., et al. (2021). Building high accuracy emulators for scientific simulations with deep neural architecture search. *Machine Learning: Science and Technology*, *3*(1), 015013. https://doi.org/10.1088/2632-2153/ac3ffa

Kochkov, D., Smith, J. A., Alieva, A., Wang, Q., Brenner, M. P., & Hoyer, S. (2021). Machine learning-accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, *118*(21). https://doi.org/10.1073/pnas.2101784118

Kraichnan, R. H. (1967). Inertial ranges in two-dimensional turbulence. *Physics of Fluids*, *10*(7), 1417–1423. https://doi.org/10.1063/1.1762301

Kraichnan, R. H. (1976). Eddy viscosity in two and three dimensions. *Journal of the Atmospheric Sciences*, *33*(8), 1521–1536. https://doi.org/10.1175/1520-0469(1976)033<1521:evitat>2.0.co;2

Krasnopolsky, V. M., Fox-Rabinovitz, M. S., & Belochitski, A. A. (2013). Using ensemble of neural networks to learn stochastic convection parameterizations for climate and numerical weather prediction models from data simulated by a cloud-resolving model. *Advances in Artificial Neural Systems*. https://doi.org/10.1155/2013/485913

Lapeyre, C. J., Misdariis, A., Cazard, N., Veynante, D., & Poinsot, T. (2019). Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates. *Combustion and Flame*, *203*, 255–264. https://doi.org/10.1016/j.combustflame.2019.02.019

Leith, C. (1996). Stochastic models of chaotic systems. *Physica D: Nonlinear Phenomena*, *98*(2–4), 481–491. https://doi.org/10.1016/0167-2789(96)00107-8

Leonard, A. (1975). Energy cascade in large-eddy simulations of turbulent fluid flows. *Advances in Geophysics*, *18*, 237–248. https://doi.org/10.1016/s0065-2687(08)60464-1

Lesieur, M., Métais, O., & Comte, P. (2005). *Large-eddy simulations of turbulence*. Cambridge University Press.

Lilly, D. K. (1992). A proposed modification of the Germano subgrid-scale closure method. *Physics of Fluids A: Fluid Dynamics*, *4*(3), 633–635. https://doi.org/10.1063/1.858280

Long, Z., Lu, Y., Ma, X., & Dong, B. (2018). PDE-Net: Learning PDES from data. In *International Conference on Machine Learning* (pp. 3208–3216).

MacArt, J. F., Sirignano, J., & Freund, J. B. (2021). Embedded training of neural-network subgrid-scale turbulence models. *Physical Review Fluids*, *6*(5), 050502. https://doi.org/10.1103/physrevfluids.6.050502

Madec, G., Bourdallé-Badie, R., Bouttier, P.-A., Bricaud, C., Bruciaferri, D., Calvert, D., et al. (2017). Nemo ocean engine.

Majda, A., & Wang, X. (2006). *Nonlinear dynamics and statistical theories for basic geophysical flows*. Cambridge University Press.

Marshall, J., Adcroft, A., Hill, C., Perelman, L., & Heisey, C. (1997). A finite-volume, incompressible Navier Stokes model for studies of the ocean on parallel computers. *Journal of Geophysical Research: Oceans*, *102*(C3), 5753–5766. https://doi.org/10.1029/96jc02775

Maulik, R., San, O., Rasheed, A., & Vedula, P. (2019). Subgrid modeling for two-dimensional turbulence using neural networks. *Journal of Fluid Mechanics*, *858*, 122–144. https://doi.org/10.1017/jfm.2018.770

Mellor, G. L. (1985). Ensemble average, turbulence closure. *Advances in Geophysics*, *28*, 345–358. https://doi.org/10.1016/s0065-2687(08)60194-6
Meyer, D., Hogan, R. J., Dueben, P. D., & Mason, S. L. (2021). Machine learning emulation of 3D cloud radiative effects. *Journal of Advances in Modeling Earth Systems*, e2021MS002550. https://doi.org/10.1029/2021MS002550
Meyers, J. (2011). Error-landscape assessment of large-eddy simulations: A review of the methodology. *Journal of Scientific Computing*, *49*(1), 65–77. https://doi.org/10.1007/s10915-010-9449-z
Mohan, A. T., Lubbers, N., Livescu, D., & Chertkov, M. (2020). Embedding hard physical constraints in convolutional neural networks for 3D turbulence. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*.
Neumann, P., Düben, P., Adamidis, P., Bauer, P., Brück, M., Kornblueh, L., et al. (2019). Assessing the scales in numerical weather and climate predictions: Will exascale be the rescue? *Philosophical Transactions of the Royal Society A*, *377*(2142), 20180148. https://doi.org/10.1098/rsta.2018.0148
Nonnenmacher, M., & Greenberg, D. S. (2021). Deep emulators for differentiation, forecasting, and parametrization in Earth science simulators. *Journal of Advances in Modeling Earth Systems*, *13*(7), e2021MS002554. https://doi.org/10.1029/2021ms002554
O'Gorman, P. A., & Dwyer, J. G. (2018). Using machine learning to parameterize moist convection: Potential for modeling of climate, climate change, and extreme events. *Journal of Advances in Modeling Earth Systems*, *10*(10), 2548–2563. https://doi.org/10.1029/2018MS001351
Ollinaho, P., Bechtold, P., Leutbecher, M., Laine, M., Solonen, A., Haario, H., & Järvinen, H. (2013). Parameter variations in prediction skill optimization at ECMWF. *Nonlinear Processes in Geophysics*, *20*(6), 1001–1010. https://doi.org/10.5194/npg-20-1001-2013
Ouala, S., Debreu, L., Pascual, A., Chapron, B., Collard, F., Gaultier, L., & Fablet, R. (2021). Learning Runge-Kutta integration schemes for ODE simulation and identification. arXiv preprint arXiv:2105.04999.
Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems* (Vol. *32*, pp. 8026–8037).
Pawar, S., San, O., Rasheed, A., & Vedula, P. (2020). A priori analysis on deep learning of subgrid-scale parameterizations for Kraichnan turbulence. *Theoretical and Computational Fluid Dynamics*, *34*(4), 429–455. https://doi.org/10.1007/s00162-019-00512-z
Pawar, S., San, O., Rasheed, A., & Vedula, P. (2022). Frame invariant neural network closures for Kraichnan turbulence. arXiv preprint arXiv:2201.02928.
Piomelli, U., Moin, P., & Ferziger, J. H. (1988). Model consistency in large eddy simulation of turbulent channel flows. *Physics of Fluids*, *31*(7), 1884–1891. https://doi.org/10.1063/1.866635
Pope, S. B. (2000). *Turbulent flows*. Cambridge University Press.
Portwood, G. D., Nadiga, B. T., Saenz, J. A., & Livescu, D. (2021). Interpreting neural network models of residual scalar flux. *Journal of Fluid Mechanics*, 907. https://doi.org/10.1017/jfm.2020.861
Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, *378*, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045
Ramadhan, A., Wagner, G., Hill, C., Campin, J.-M., Churavy, V., Besard, T., et al. (2020). Oceananigans.jl: Fast and friendly geophysical fluid dynamics on GPUs. *Journal of Open Source Software*, *5*(53). https://doi.org/10.21105/joss.02018
Rasp, S., Pritchard, M. S., & Gentine, P. (2018). Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, *115*(39), 9684–9689. https://doi.org/10.1073/pnas.1810286115
Schneider, T., Lan, S., Stuart, A., & Teixeira, J. (2017). Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations. *Geophysical Research Letters*, *44*(24), 12–396. https://doi.org/10.1002/2017gl076101
Schneider, T., Teixeira, J., Bretherton, C. S., Brient, F., Pressel, K. G., Schär, C., & Siebesma, A. P. (2017). Climate goals and computing the future of clouds. *Nature Climate Change*, *7*(1), 3–5. https://doi.org/10.1038/nclimate3190
Schultz, M., Betancourt, C., Gong, B., Kleinert, F., Langguth, M., Leufen, L., et al. (2021). Can deep learning beat numerical weather prediction? *Philosophical Transactions of the Royal Society A*, *379*(2194), 20200097. https://doi.org/10.1098/rsta.2020.0097
Seifert, A., & Rasp, S. (2020). Potential and limitations of machine learning for modeling warm-rain cloud microphysical processes. *Journal of Advances in Modeling Earth Systems*, *12*(12), e2020MS002301. https://doi.org/10.1029/2020ms002301
Sirignano, J., MacArt, J. F., & Freund, J. B. (2020). DPM: A deep learning PDE augmentation method with application to large-eddy simulation. *Journal of Computational Physics*, *423*, 109811. https://doi.org/10.1016/j.jcp.2020.109811
Sirignano, J., & Spiliopoulos, K. (2018). DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, *375*, 1339–1364. https://doi.org/10.1016/j.jcp.2018.08.029
Smagorinsky, J. (1963). General circulation experiments with the primitive equations: I. The basic experiment. *Monthly Weather Review*, *91*(3), 99–164. https://doi.org/10.1175/1520-0493(1963)091<0099:gcewtp>2.3.co;2
Sridhar, A., Tissaoui, Y., Marras, S., Shen, Z., Kawczynski, C., Byrne, S., et al. (2021). Large-eddy simulations with ClimateMachine v0. 2.0: A new open-source code for atmospheric simulations on GPUS and CPUS. *Geoscientific Model Development Discussions*, *15*, 6259–6284.
Stachenfeld, K., Fielding, D. B., Kochkov, D., Cranmer, M., Pfaff, T., Godwin, J., et al. (2021). Learned coarse models for efficient turbulence simulation. arXiv preprint arXiv:2112.15275.
Stensrud, D. J. (2009). *Parameterization schemes: Keys to understanding numerical weather prediction models*. Cambridge University Press.
Thuburn, J., Kent, J., & Wood, N. (2014). Cascades, backscatter, and conservation in numerical models of two-dimensional turbulence. *Quarterly Journal of the Royal Meteorological Society*, *140*(679), 626–638. https://doi.org/10.1002/qj.2166
Thuerey, N., Holl, P., Mueller, M., Schnell, P., Trost, F., & Um, K. (2021). *Physics-based deep learning*. WWW. Retrieved from https://physicsbaseddeeplearning.org
Ukkonen, P., Pincus, R., Hogan, R. J., Pagh Nielsen, K., & Kaas, E. (2020). Accelerating radiation computations for dynamical models with targeted machine learning and code optimization. *Journal of Advances in Modeling Earth Systems*, *12*(12), e2020MS002226. https://doi.org/10.1029/2020ms002226
Um, K., Brand, R., Fei, Y. R., Holl, P., & Thuerey, N. (2020). Solver-in-the-loop: Learning from differentiable physics to interact with iterative PDE-solvers. *Advances in Neural Information Processing Systems*, *33*, 6111–6122.
Vidard, A., Bouttier, P.-A., & Vigilant, F. (2015). NEMOTAM: Tangent and adjoint models for the ocean modeling platform NEMO. *Geoscientific Model Development*, *8*(4), 1245–1257. https://doi.org/10.5194/gmd-8-1245-2015
Vinuesa, R., & Brunton, S. L. (2021). The potential of machine learning to enhance computational fluid dynamics. arXiv preprint arXiv:2110.02085.
Vollant, A., Balarac, G., & Corre, C. (2017). Subgrid-scale scalar flux modeling based on optimal estimation theory and machine-learning procedures. *Journal of Turbulence*, *18*(9), 854–878. https://doi.org/10.1080/14685248.2017.1334907
Weyn, J. A., Durran, D. R., Caruana, R., & Cresswell-Clay, N. (2021). Sub-seasonal forecasting with a large ensemble of deep-learning weather prediction models. *Journal of Advances in Modeling Earth Systems*, *13*(7), e2021MS002502. https://doi.org/10.1029/2021ms002502

Wunsch, C., & Heimbach, P. (2013). Dynamically and kinematically consistent global ocean circulation and ice state estimates. *International Geophysics*, *103*, 553–579. https://doi.org/10.1016/b978-0-12-391851-2.00021-0

Xie, C., Wang, J., & Weinan, E. (2020). Modeling subgrid-scale forces by spatial artificial neural networks in large eddy simulation of turbulence. *Physical Review Fluids*, *5*(5), 054606. https://doi.org/10.1103/physrevfluids.5.054606

Yuan, Z., Xie, C., & Wang, J. (2020). Deconvolutional artificial neural network models for large eddy simulation of turbulence. *Physics of Fluids*, *32*(11), 115106. https://doi.org/10.1063/5.0027146

Zanna, L., & Bolton, T. (2020). Data-driven equation discovery of ocean mesoscale closures. *Geophysical Research Letters*, *47*(17), e2020GL088376. https://doi.org/10.1029/2020gl088376

Zanna, L., & Bolton, T. (2021). Deep learning of unresolved turbulent ocean processes in climate models. *Deep Learning for the Earth Sciences: A Comprehensive Approach to Remote Sensing, Climate Science, and Geosciences*, 298–306. https://doi.org/10.1002/9781119646181.ch20

Zhou, Z., He, G., Wang, S., & Jin, G. (2019). Subgrid-scale model for large-eddy simulation of isotropic turbulent flows using an artificial neural network. *Computers & Fluids*, *195*, 104319. https://doi.org/10.1016/j.compfluid.2019.104319