Why and how to learn end-to-end subgrid closures for atmosphere and ocean models?

Julien Le Sommer - CNRS, Grenoble

Hugo Frezat, Ronan Fablet, Guillaume Balarac,
Redouane Lguensat and Anastasia Gorbunova

ECMWF ML workshop, virtual
29th March 2022

Why and how to learn end-to-end subgrid closures for atmosphere and ocean models?

Julien Le Sommer - CNRS, Grenoble

Hugo Frezat, Ronan Fablet, Guillaume Balarac,
Redouane Lguensat and Anastasia Gorbunova

ECMWF ML workshop, virtual
29th March 2022

Machine learning for subgrid closures

Dynamical system

$$\partial_t \mathbf{x} + \mathscr{L}\,\mathbf{x} + \mathscr{N}(\mathbf{x}) = 0$$

Resolved system

$$\partial_t \widetilde{\mathbf{x}} + \mathscr{L}\,\widetilde{\mathbf{x}} + \mathscr{N}(\widetilde{\mathbf{x}}) = \mathscr{N}(\widetilde{\mathbf{x}}) - \widetilde{\mathscr{N}(\mathbf{x})}$$

Subgrid term

$$\mathscr{M}(\widetilde{\mathbf{x}}) \simeq \mathscr{N}(\widetilde{\mathbf{x}}) - \widetilde{\mathscr{N}(\mathbf{x})}$$

Parameterizations

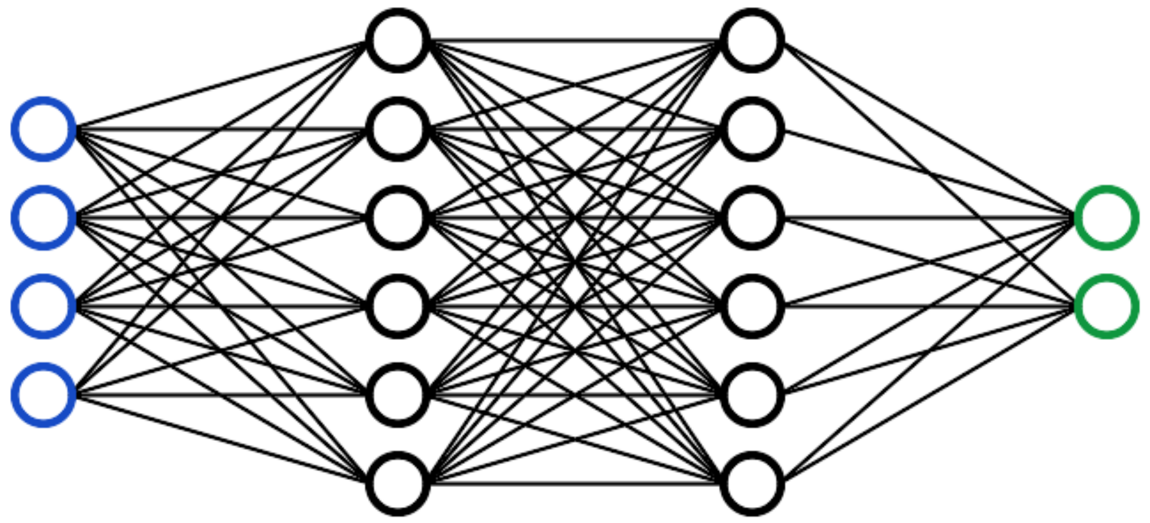
atmosphere : clouds, convection, gravity waves

ocean : mixing, macro-turbulence, boundary layer

Non-trivial:

anisotropy, numerics, processes, scale-dependance

Machine learning



calibration, acceleration, design

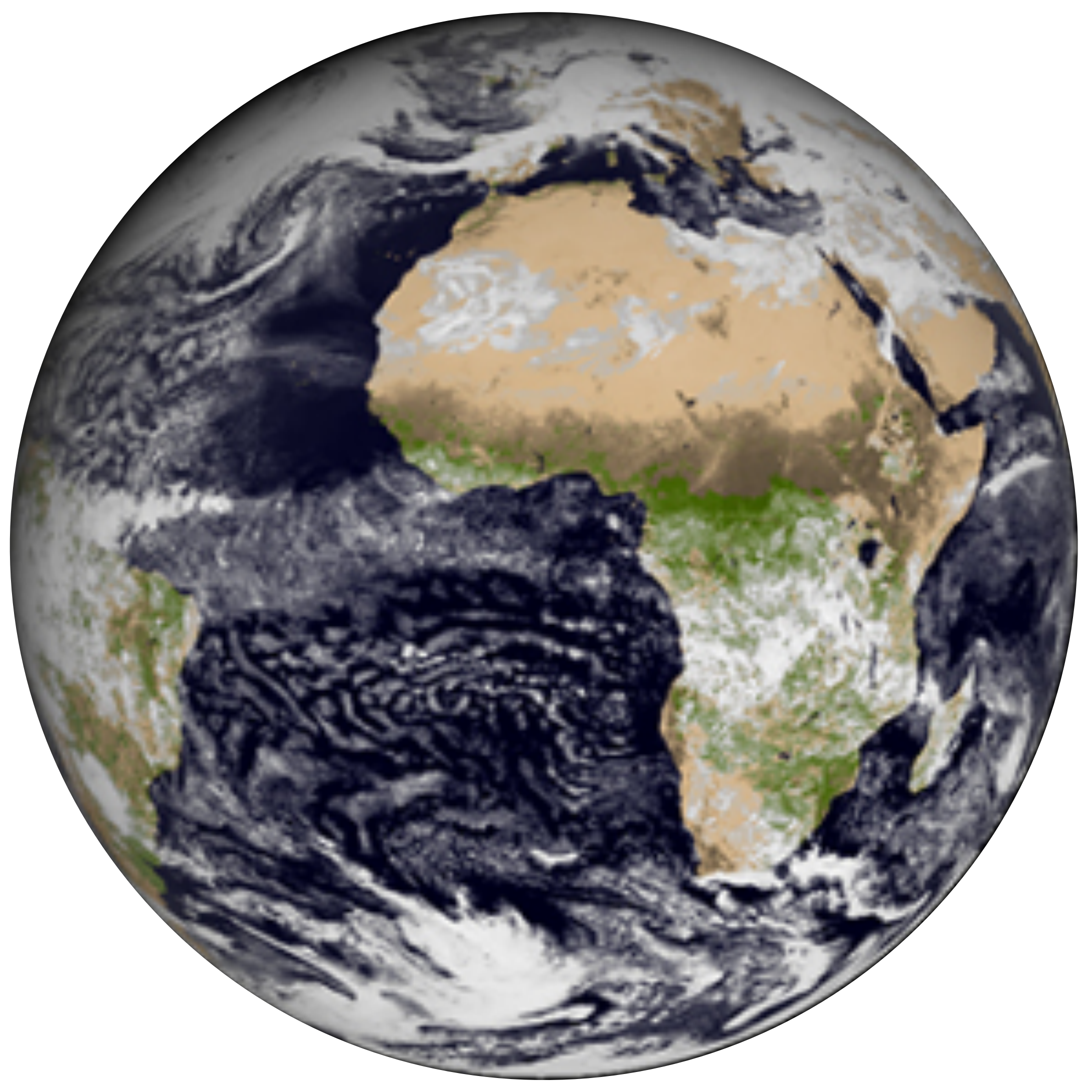
Machine learning for subgrid closures

1. Define an **averaging procedure**

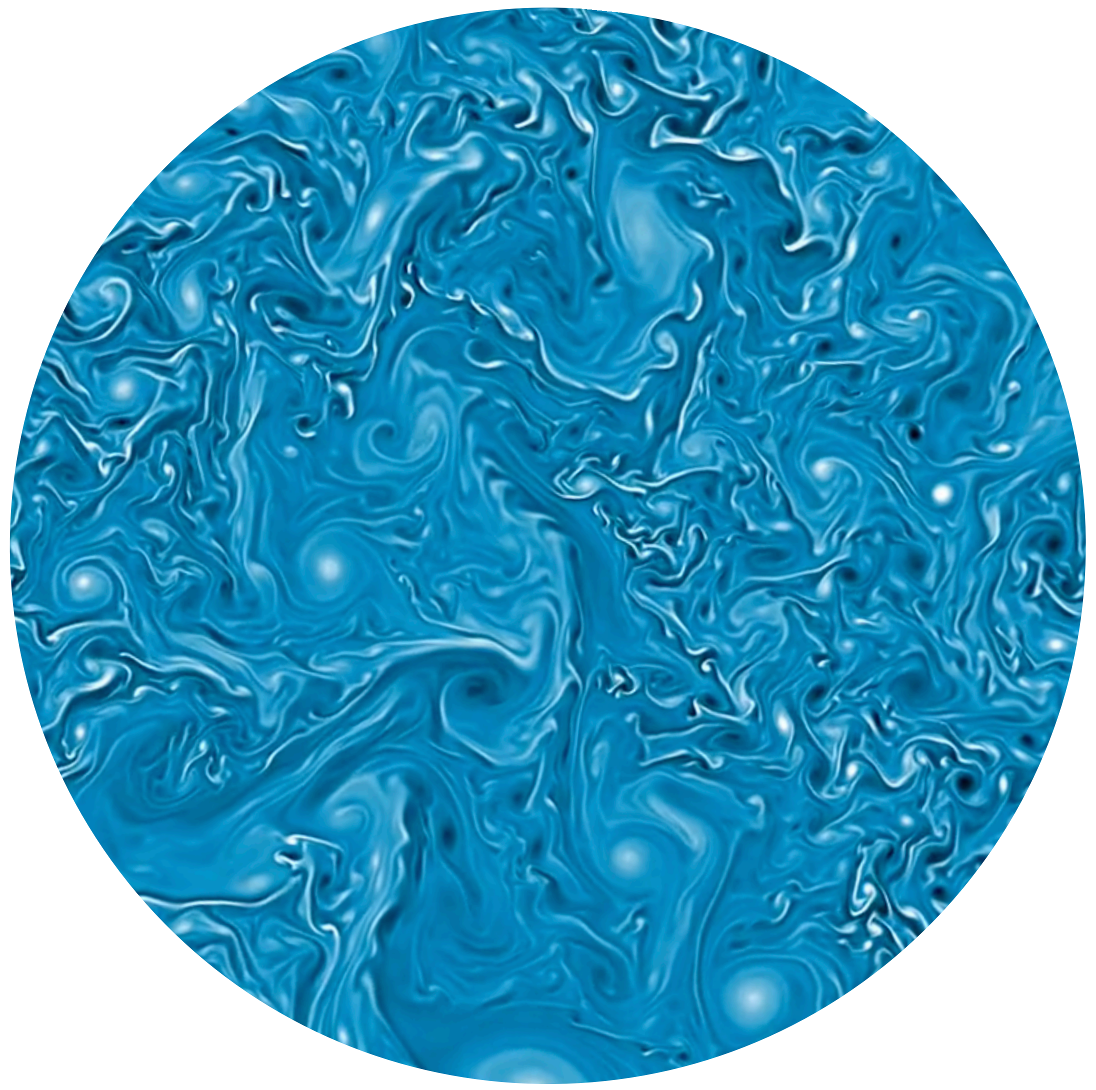
assume $\widetilde{\mathbf{x}} \sim \overline{\mathbf{x}}$

2. use a trustworthy **hi-res model**

database $\left(\overline{\mathbf{x}}, \mathcal{N}(\overline{\mathbf{x}}) - \overline{\mathcal{N}(\mathbf{x})}\right)$



cloud resolving



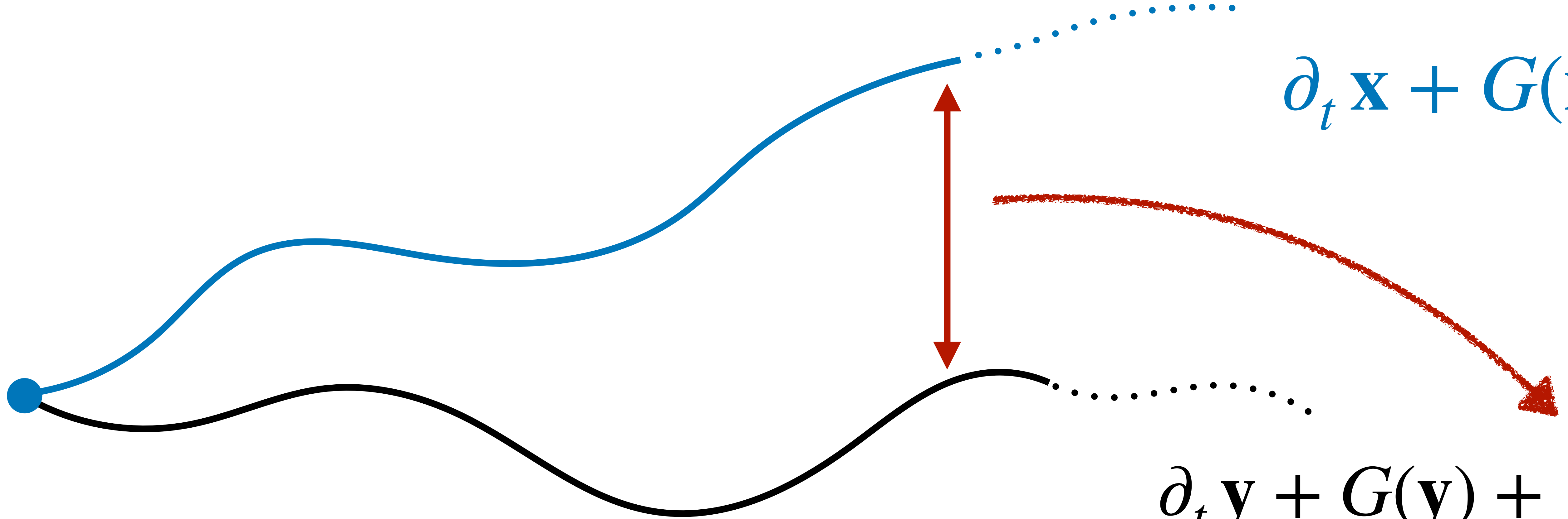
sub-mesoscale permitting

3. **supervised learning** problem

learn $\overline{\mathbf{x}} \to \mathcal{N}(\overline{\mathbf{x}}) - \overline{\mathcal{N}(\mathbf{x})}$

Subgrid term

$$\mathcal{M}(\widetilde{\mathbf{x}}) \simeq \mathcal{N}(\widetilde{\mathbf{x}}) - \widetilde{\mathcal{N}(\mathbf{x})}$$

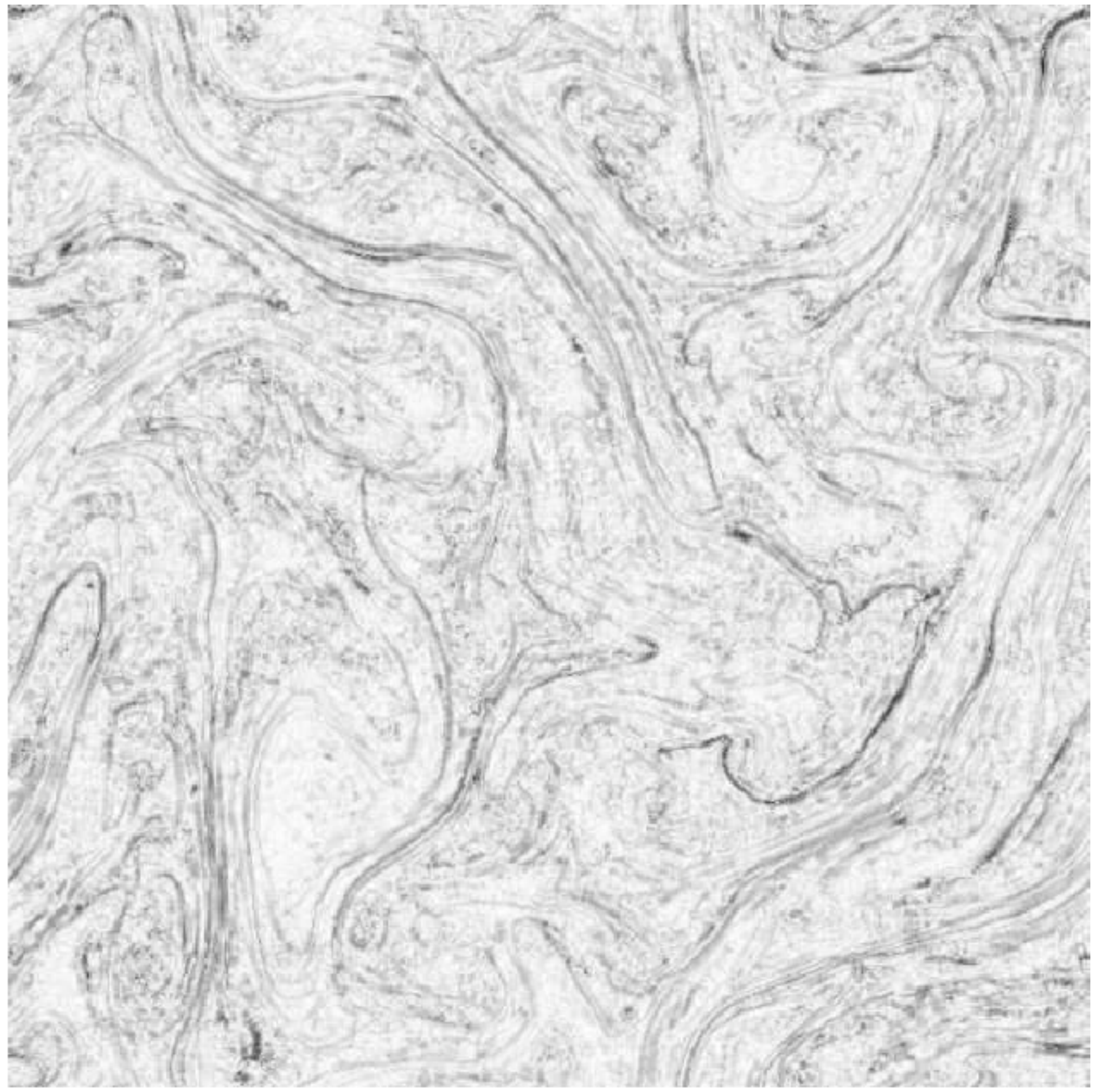


$$\partial_t \mathbf{x} + G(\mathbf{x}) = f$$

$$\partial_t \mathbf{y} + G(\mathbf{y}) + \ ? \ = f$$

A priori vs a posteriori skill

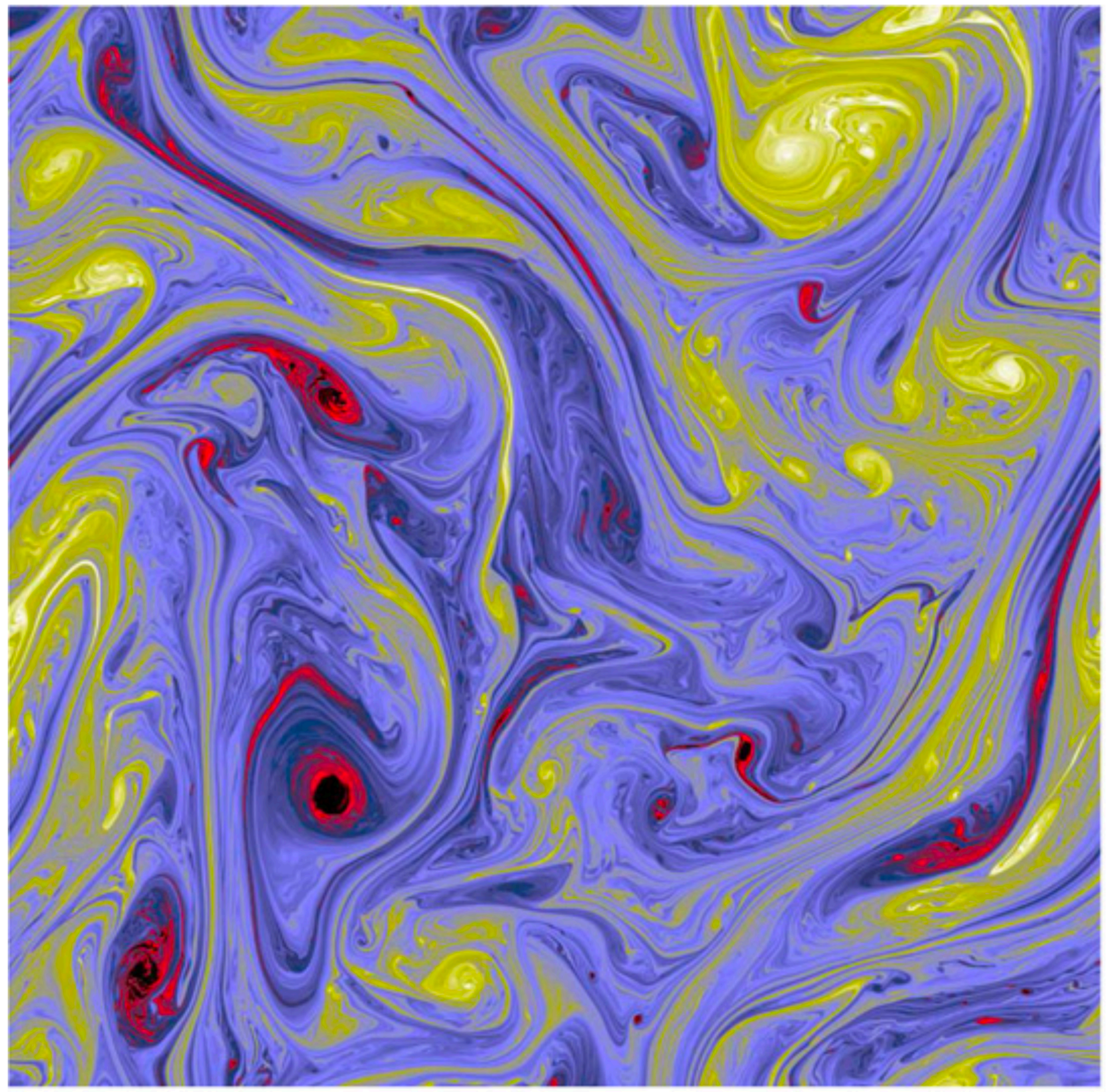
in Large Eddy Simulation



<u>a priori</u> skill :

ability to predict the unknown term

at fixed time t



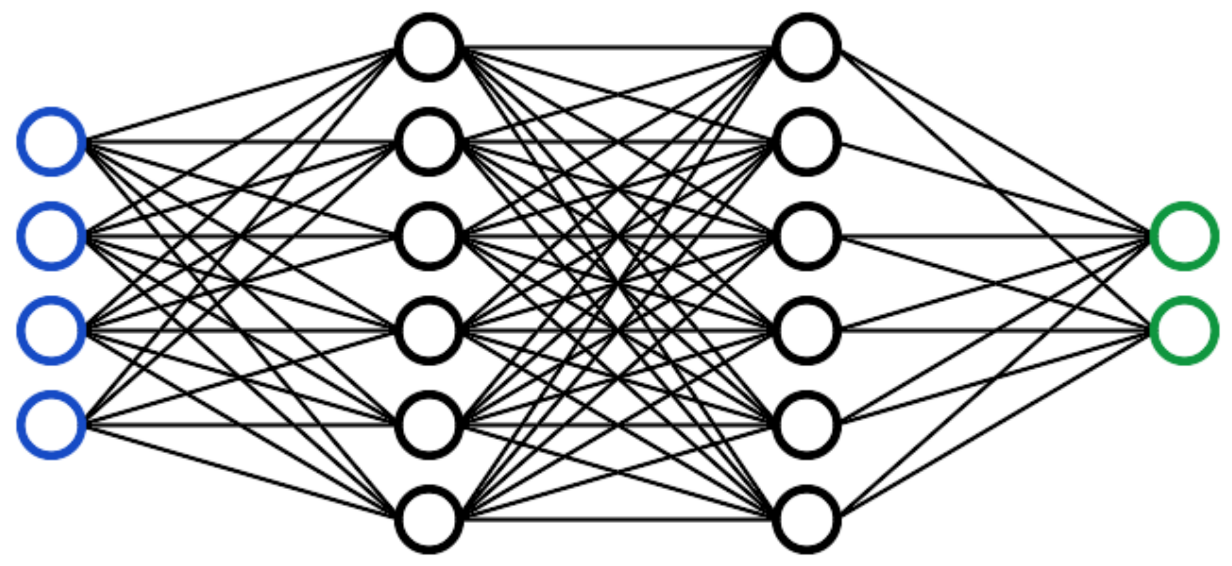
<u>a posteriori</u> skill :

impact on the model solution

along a trajectory

Two learning strategies

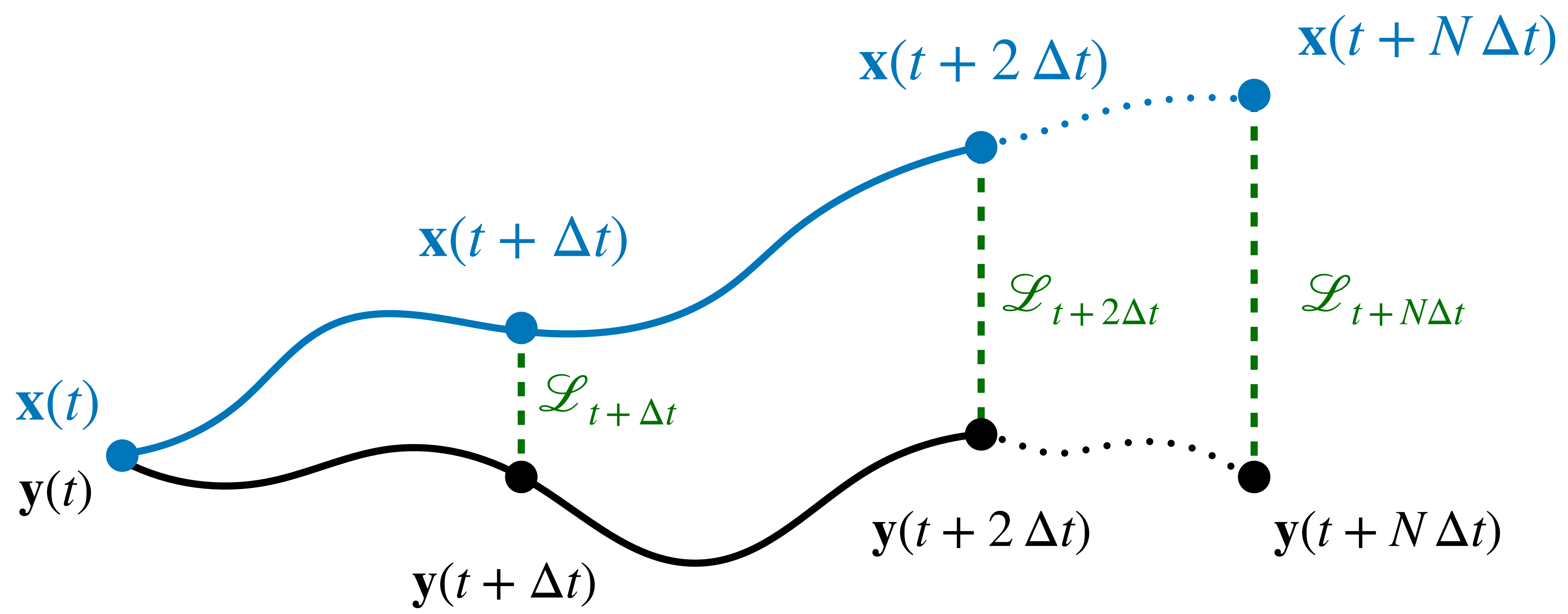
a priori learning



mapping $\bar{\mathbf{x}} \rightarrow \overline{\mathcal{N}(\mathbf{x})}$

at fixed time t

a posteriori learning



$$\partial_t \mathbf{y} + G(\mathbf{y}) + \mathscr{M}_{NN}(\mathbf{y}) = f$$

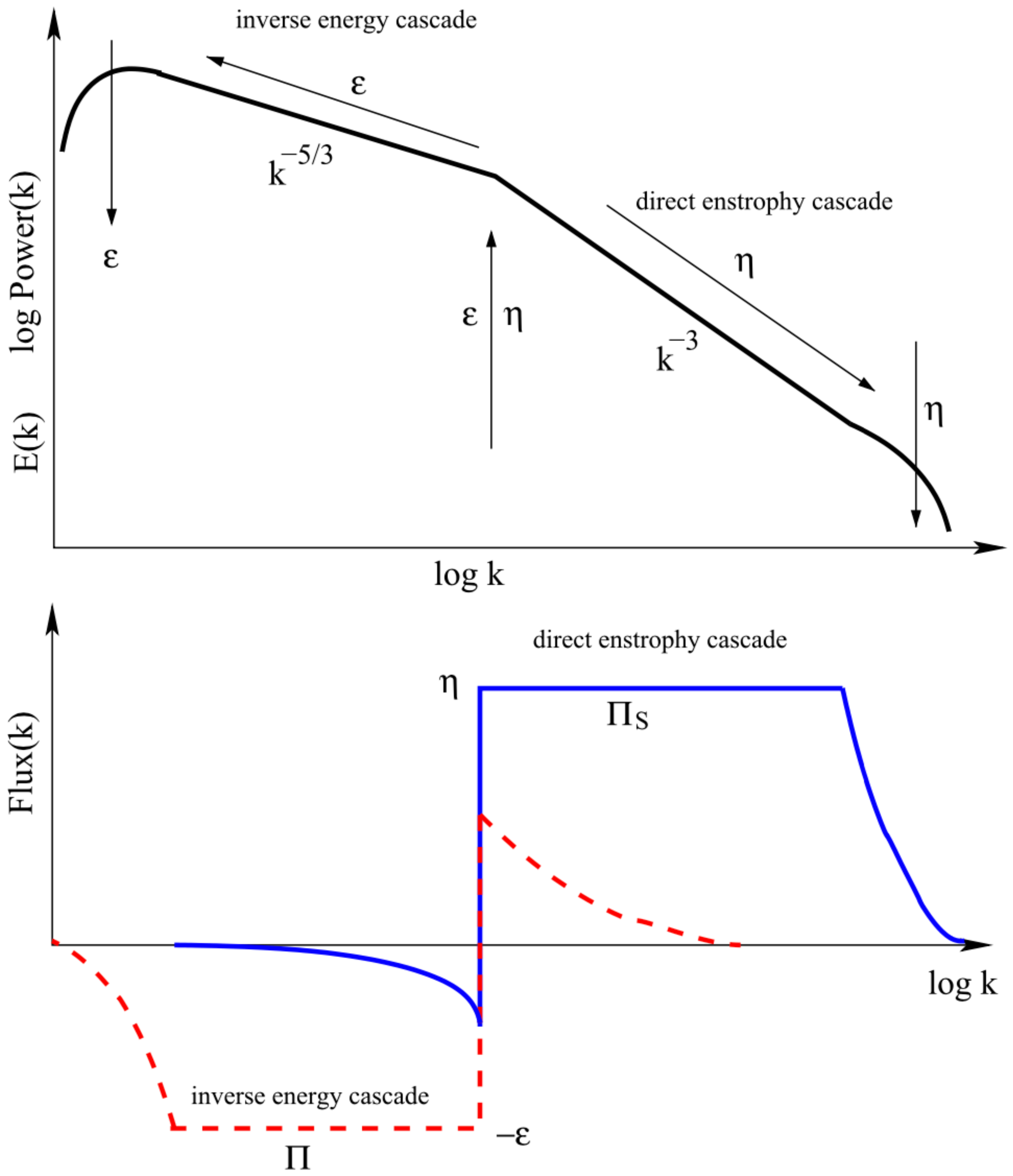
along a trajectory

Performance ? Stability ? Ability to generalise ?

Problem setting

Quasi-geostrophic turbulence

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



$$\omega = \nabla^2 \psi$$

vorticity

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

velocity

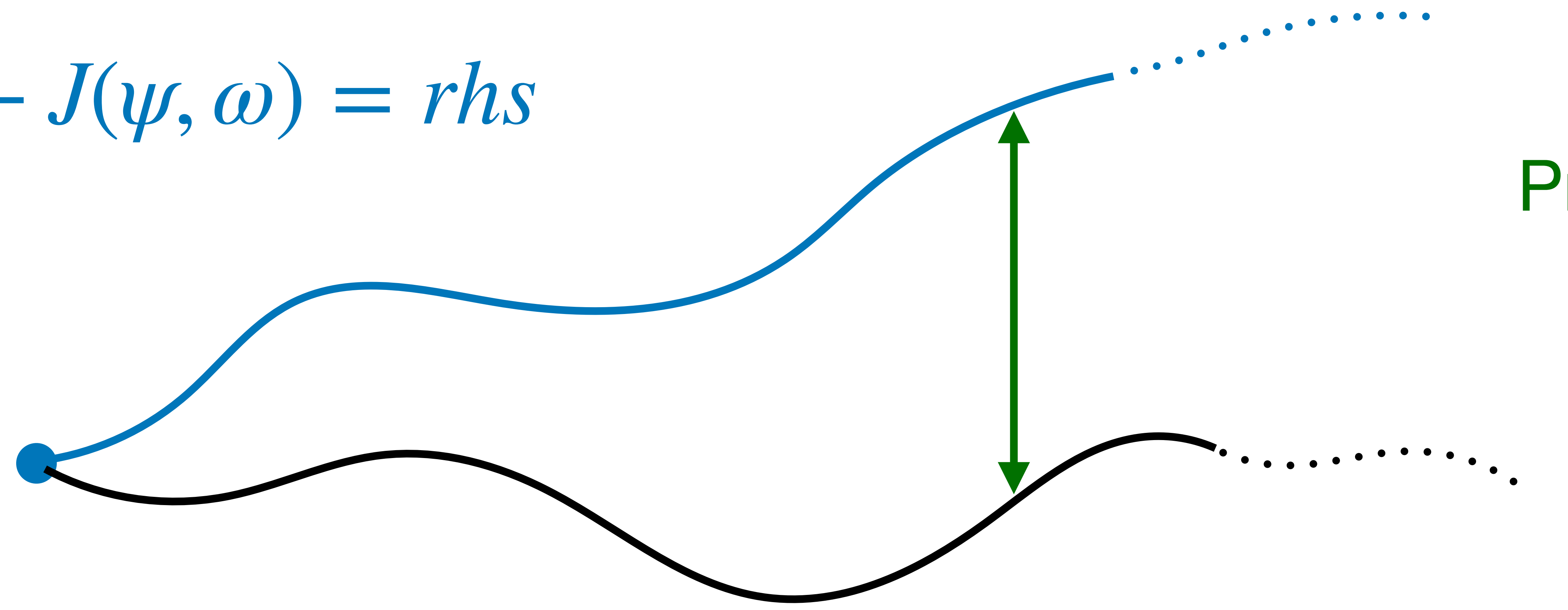
Energy $$E = \frac{1}{2}\int \mathbf{u}^2 \mathrm{d}r$$

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

Invariants

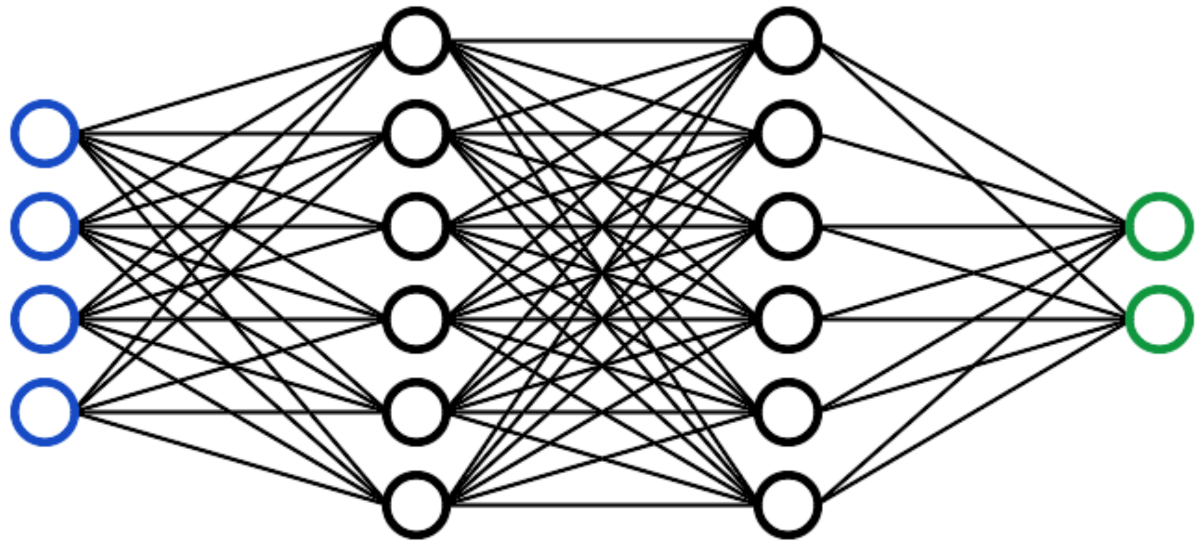
The subgrid closure problem

$\partial_t \omega + J(\psi, \omega) = rhs$



Projection $\mathcal{T}(\omega) = \overline{\omega} := \int \omega(\mathbf{x}')G(\mathbf{x}-\mathbf{x}')\mathrm{d}\mathbf{x}'$

$\partial_t \bar{\omega} + J(\bar{\psi}, \bar{\omega}) = rhs + \mathcal{M}_\theta^{NN}(\bar{\psi}, \bar{\omega})$

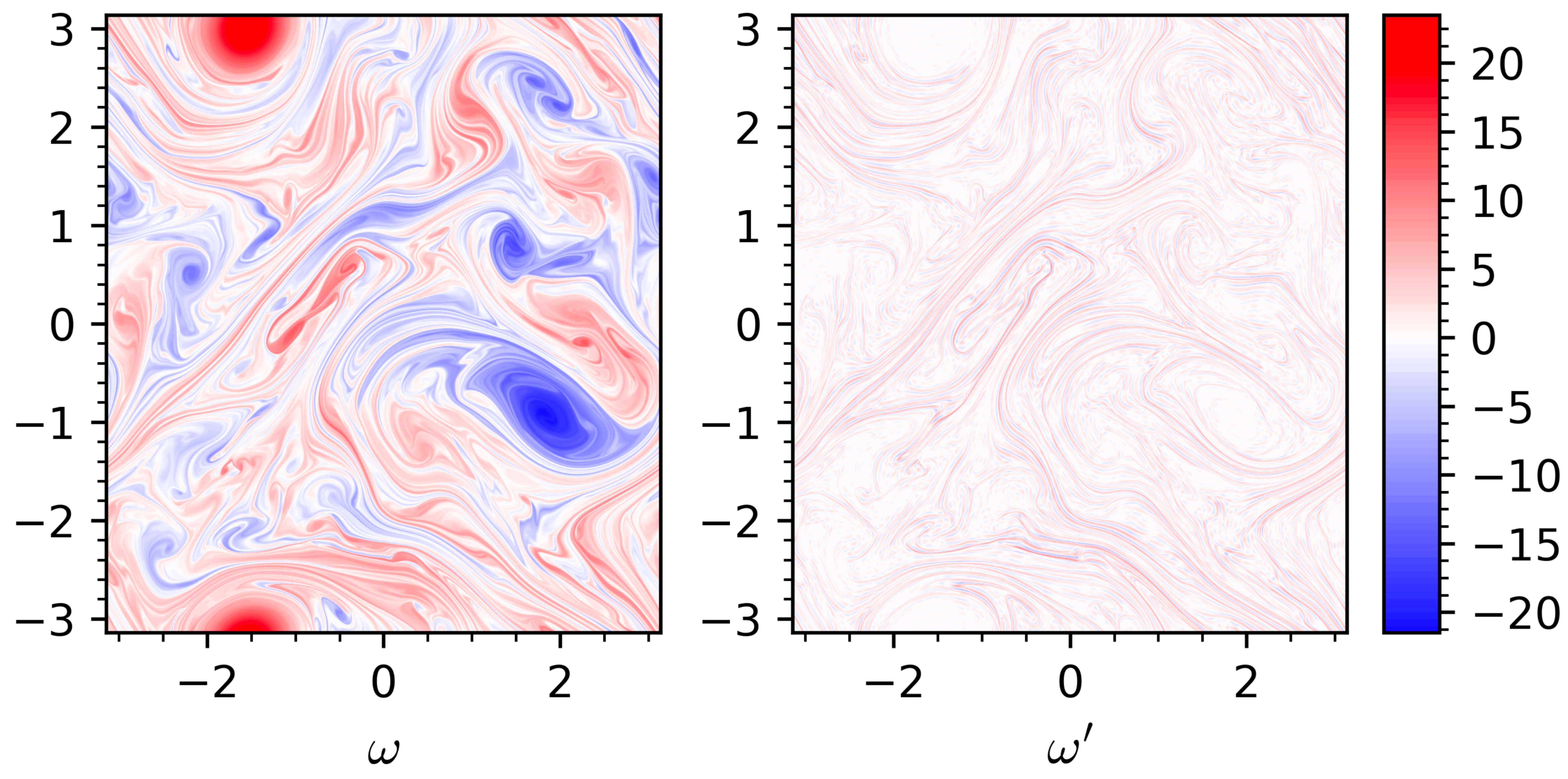


$$\partial_t \bar{\omega} + J(\bar{\psi}, \bar{\omega}) = \nu \nabla^2 \bar{\omega} - \mu \bar{\omega} - \beta \partial_x \bar{\psi} + \bar{F} + R(\psi, \omega)$$

Reduced equation

$$R(\psi, \omega) = \nabla \cdot (\bar{\mathbf{u}}\,\bar{\omega} - \overline{\mathbf{u}\,\omega})$$

Numerical solver and flow configurations



Spectral solver in PYTORCH

- **forced turbulence** : equilibrium
- **decay** : range of regimes
- **beta-plane** : jet-like turbulence

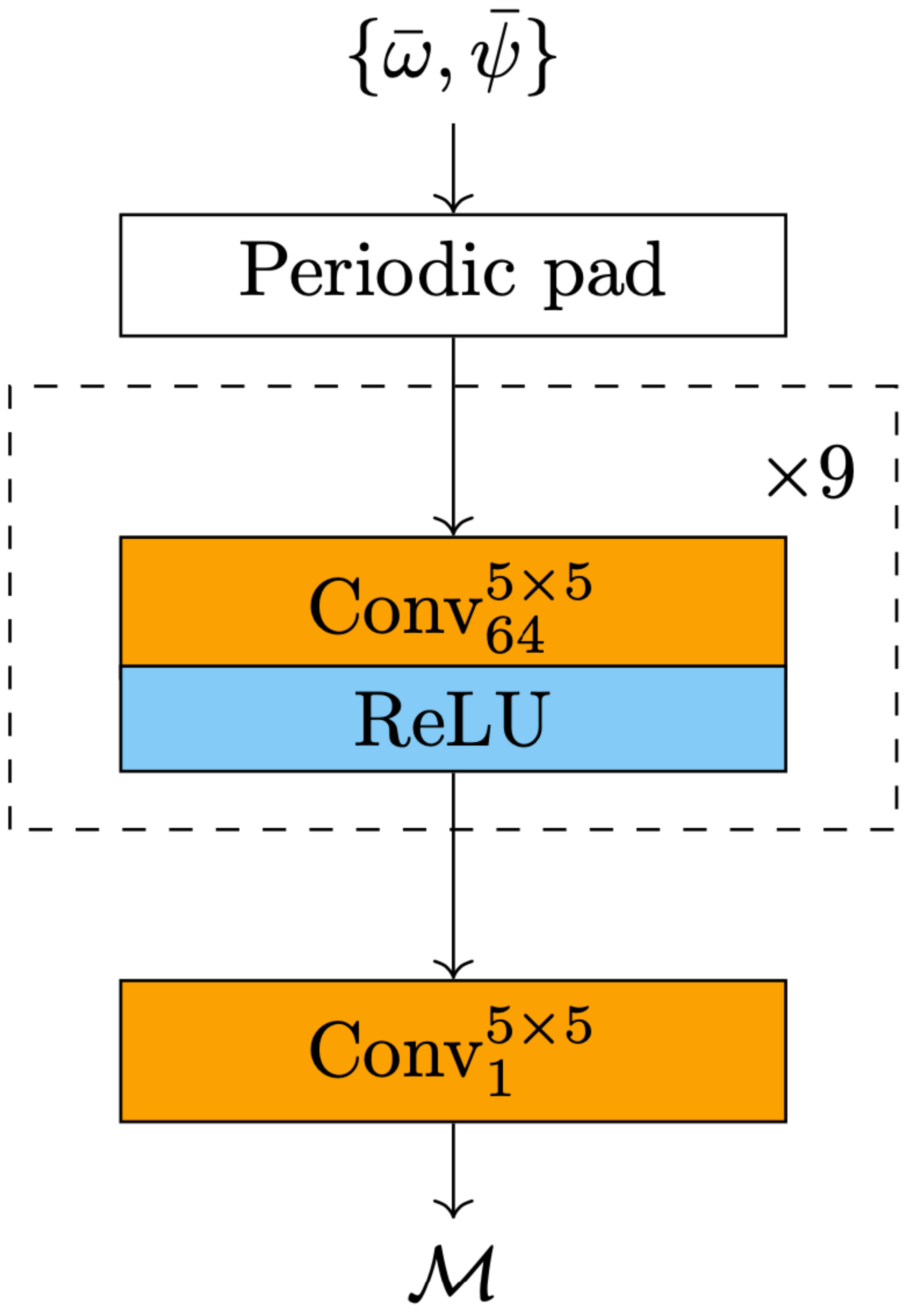
(Graham and Ringler 2013 ; Maulik et al. 2019)

Gaussian $G_\delta(k) = \exp\left(-\frac{k^2\Delta_{\text{true}}^2}{24}\right)$

Cut-off $G_\delta(k) = 0,\ \forall k > \pi\Delta_{\text{reduced}}^{-1}$

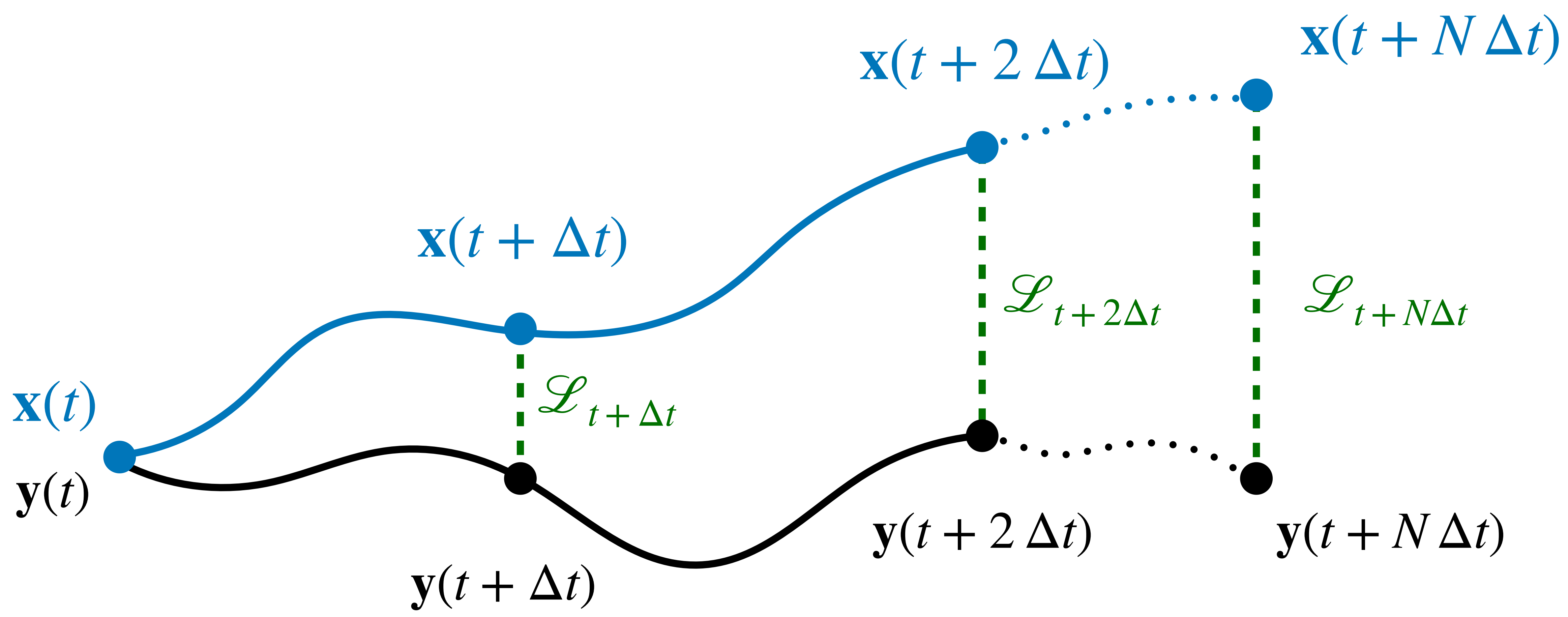
Name	$N_x \times N_y$	$L_x \times L_y$ km	Δt s	μ m^{-1}	ν $\mathrm{m}^2\mathrm{s}^{-1}$	β $\mathrm{m}^{-1}\mathrm{s}^{-1}$	Re
Decay	2048×2048	$10^4 \times 10^4$	120	0	67.0	0	32×10^3
Forced	2048×2048	$10^4 \times 10^4$	120	1.25×10^{-8}	22.0	0	22×10^4
Beta-plane	2048×2048	$10^4 \times 10^4$	120	1.25×10^{-8}	22.0	1.14×10^{-11}	34×10^4

Architecture and training



Loss for a priori training

$$\mathscr{L}_{\mathrm{prio}}(\mathscr{M}) := \frac{1}{S}\sum_{i=1}^{S}\left(R(\psi,\omega)_i - \mathscr{M}(\bar{\psi}_i, \bar{\omega}_i)\right)^2$$



Loss for a posteriori training

$$\mathscr{L}_{\mathrm{post}}(\mathscr{M}) := \frac{1}{N}\sum_{i=1}^{N}\left(\mathscr{T}(\omega(i\Delta t)) - \bar{\omega}(i\Delta t)\right)^2$$

Baseline empirical closures

$$\mathscr{M}_{\mathrm{P}}(\bar{\psi}, \bar{\omega}) = \nu_e \nabla^2 \bar{\omega}$$ eddy viscosity

Smagorinsky model

$$\nu_e = (c_{\mathrm{S}}\Delta)^2 |\bar{S}|$$

Strain rate

Leith model

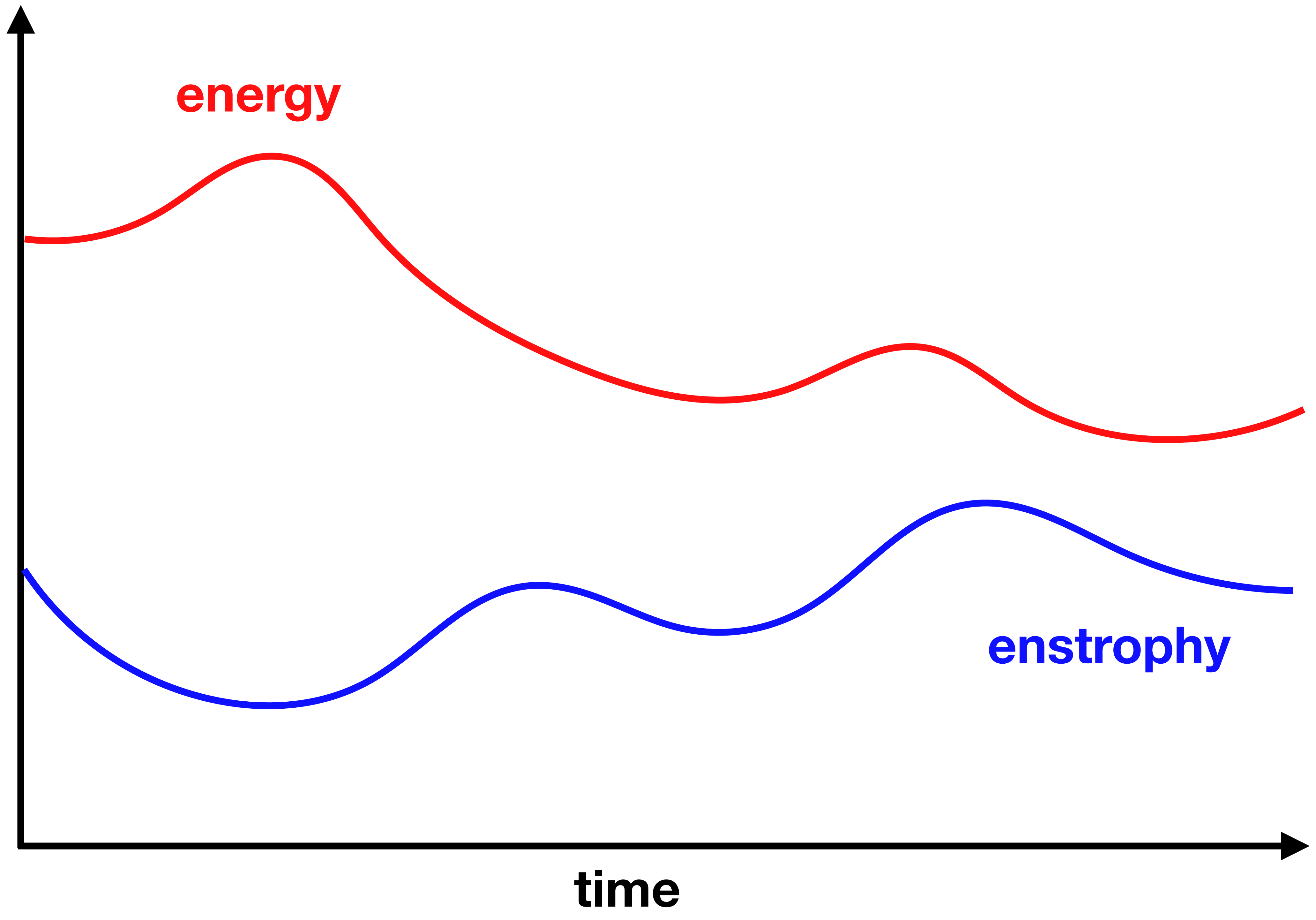
$$\nu_e = (c_{\mathrm{L}}\Delta)^3 |\nabla \bar{\omega}|$$

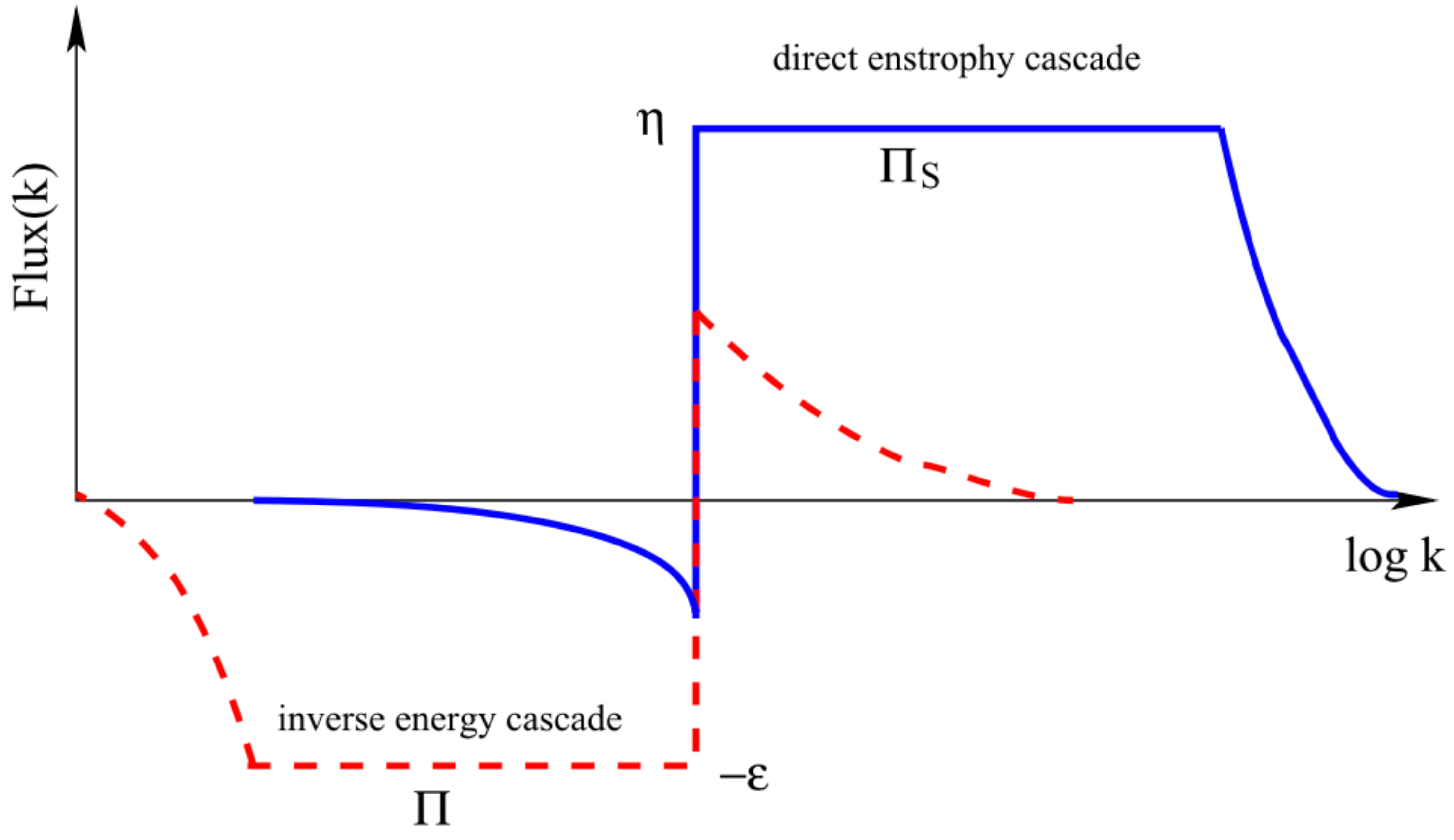
Vorticity gradients

NB with dynamic procedure (Germano 1993)

Performance metrics

Graham and Ringler (2013)





Domain-integrated energy/enstrophy

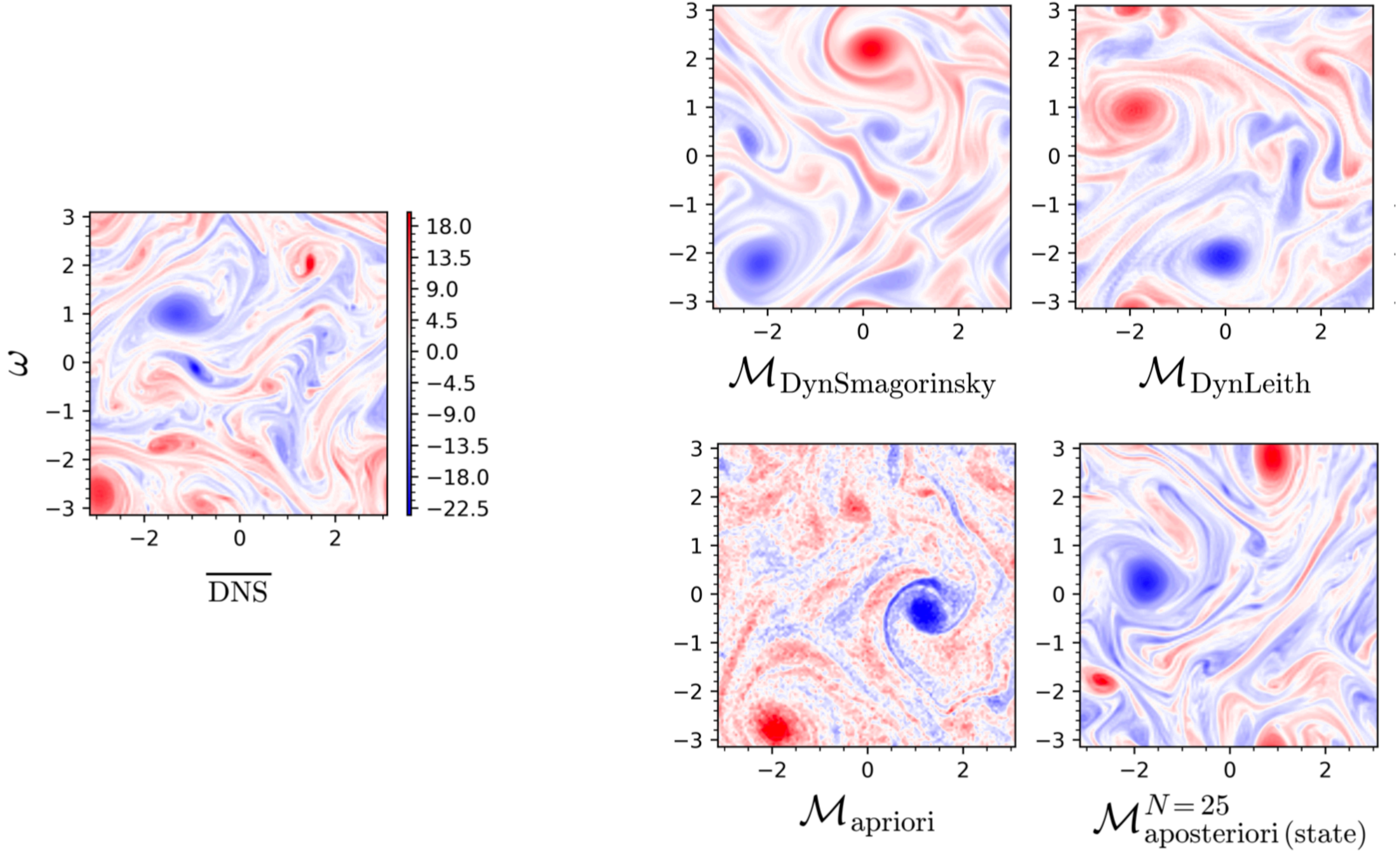
short term / forecast-like

Cross-scale fluxes at equilibrium

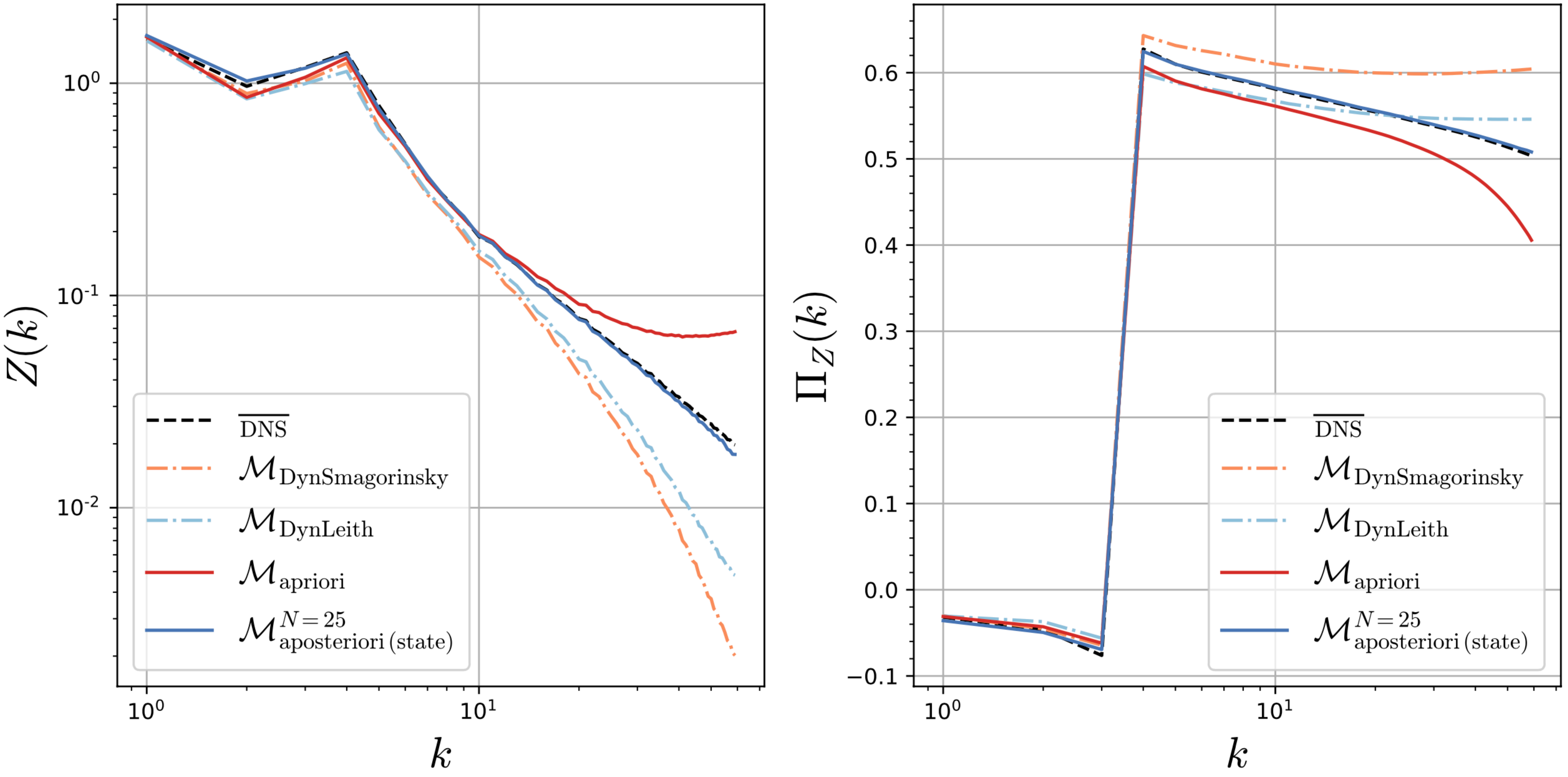
long term /climate-like

RESULTS

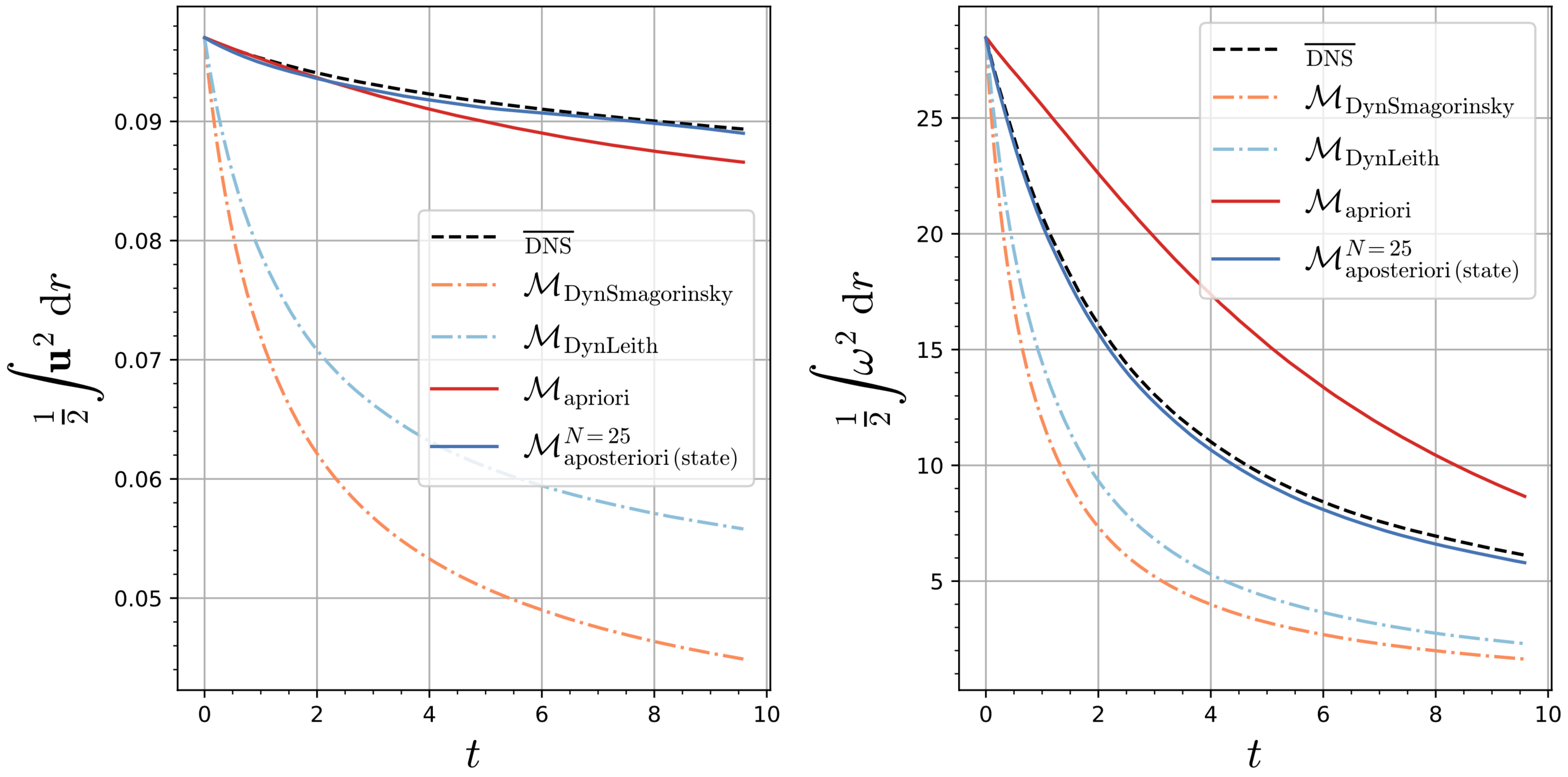
Forced turbulence



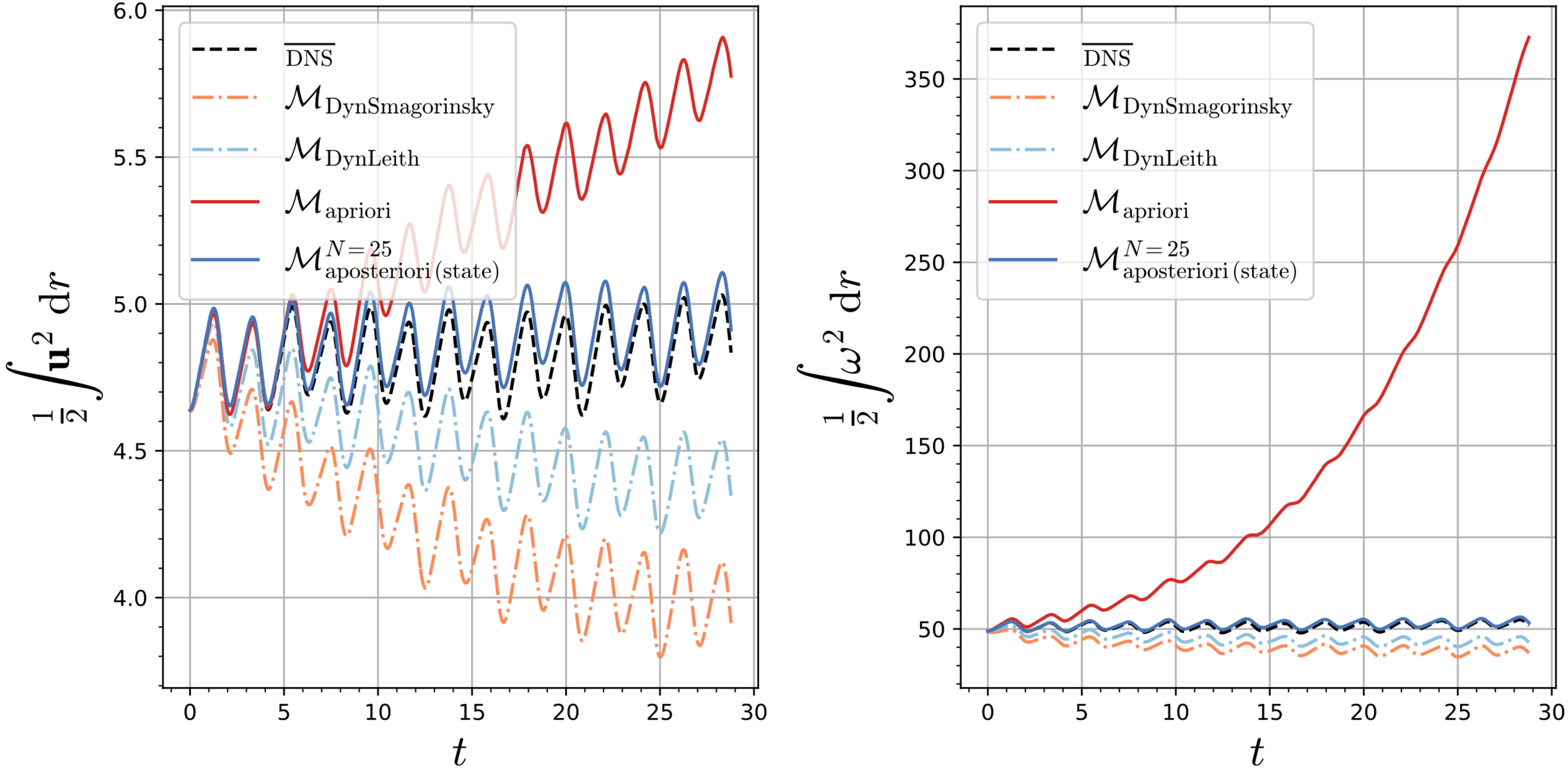
Forced turbulence



Decaying turbulence



beta-plane turbulence



Relative performance

$\rho_{R,\mathcal{M}}$	Decay		Forced		Beta-plane	
	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
$\mathcal{M}_{\text{apriori}}$	0.75	**0.90**	**0.82**	**0.95**	**0.82**	**0.96**
$\mathcal{M}_{\text{aposteriori (states)}}$	**0.77**	0.57	0.45	0.29	0.48	0.21

A priori metric

correlation with SGS

(larger is better)

$L^2(\Pi_Z^R - \Pi_Z^{\mathcal{M}})$	Decay		Forced		Beta-plane	
	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
$\mathcal{M}_{\text{apriori}}$	0.74	0.60	0.36	0.40	8.73	0.26
$\mathcal{M}_{\text{aposteriori (states)}}$	**0.13**	**0.09**	**0.02**	**0.02**	**0.30**	**0.05**

A posteriori metric

enstrophy flux

(smaller is better)

Summary (so far)

- Subgrid closures can be learned end-to-end with a posteriori criteria involving model integration over several time-steps.

- Application of end-to-end learning to quasi-geostrophic turbulent flows solves numerical stability issues related to energy backscatter.

- Learned closures outperform existing baselines for various evaluation metrics (and generalize to different flow configurations)

THOUGHTS

Leveraging deep differentiable emulators

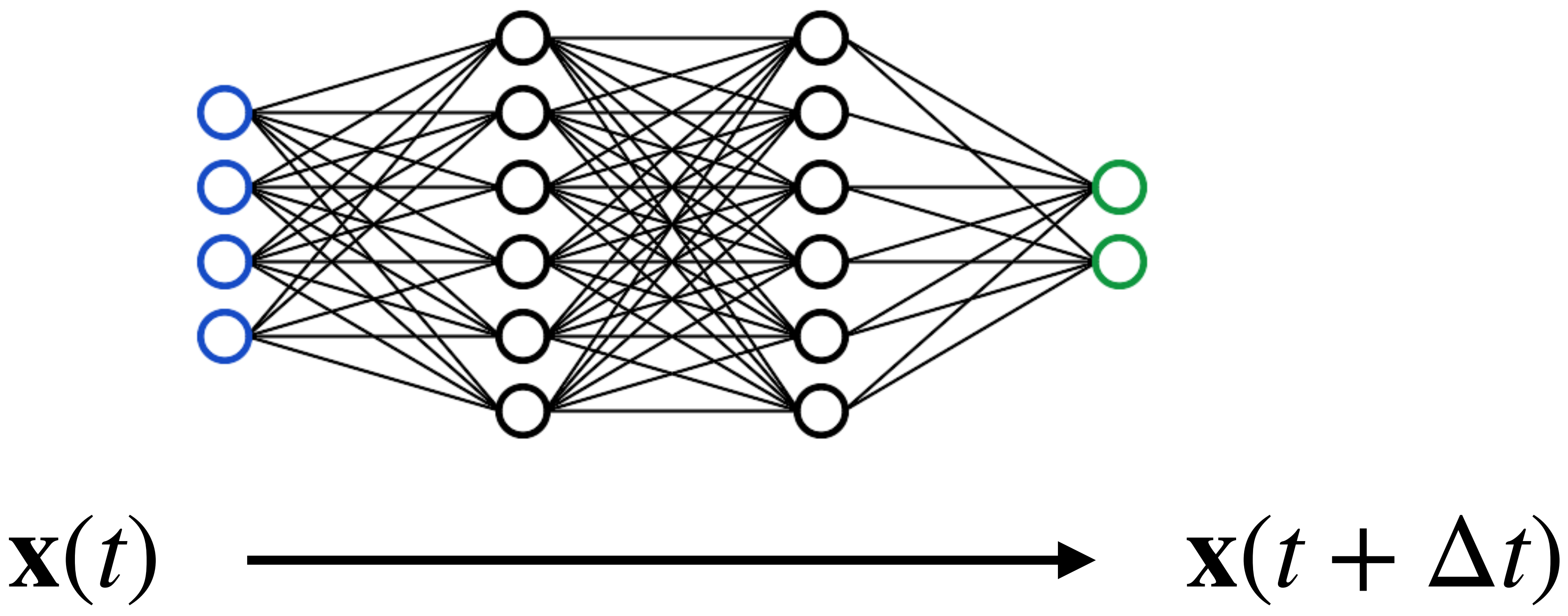
Option #1

Rewrite legacy codes

Automatic differentiation

Ex : Oceananigans.jl (Julia) Veros (Jax)

Option #2



Emulate legacy codes

Nonnenmacher and Greenberg 2021
Hatfield et al. 2021, Kasim et al. 2022

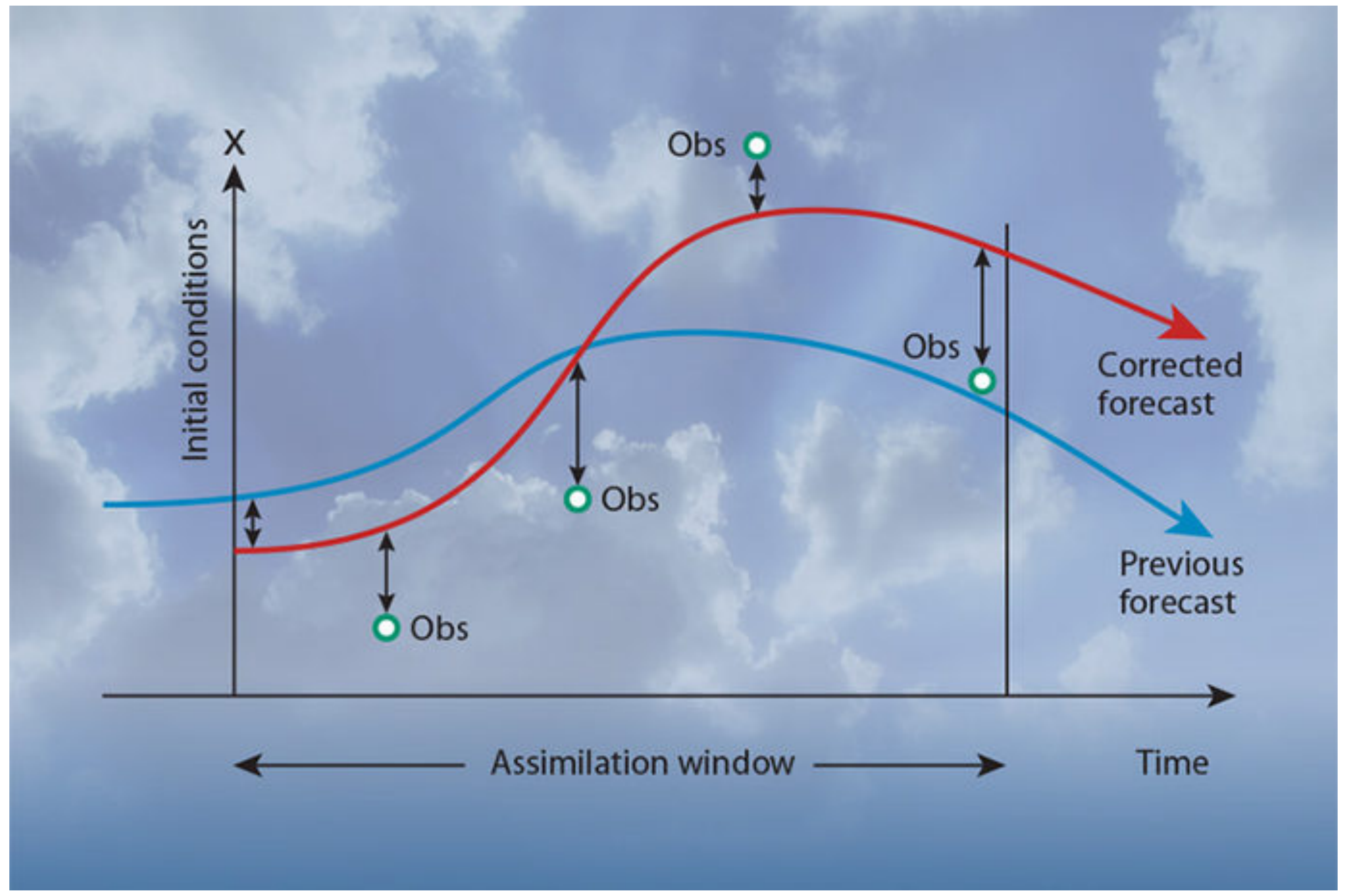
End-to-end training and 4D-Var

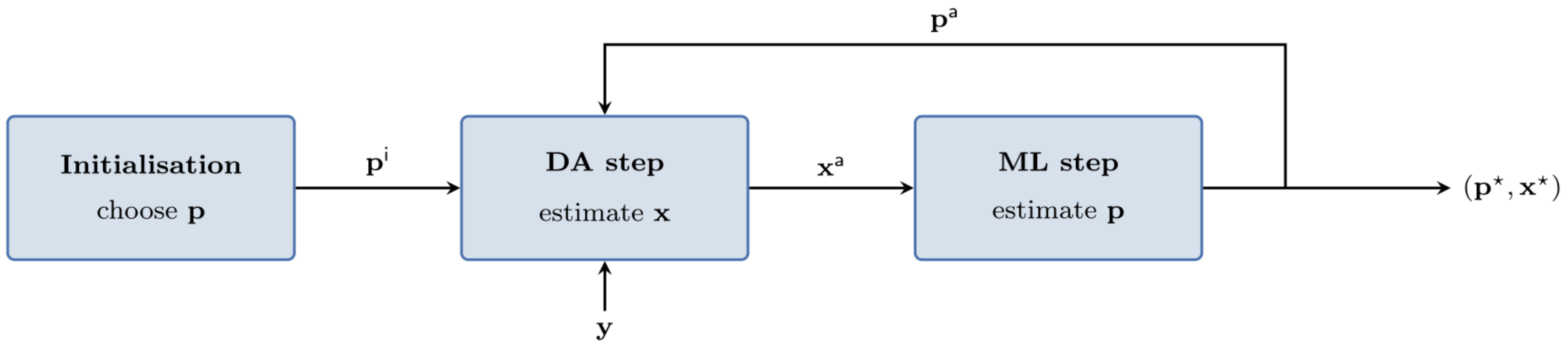
End-to-end learning

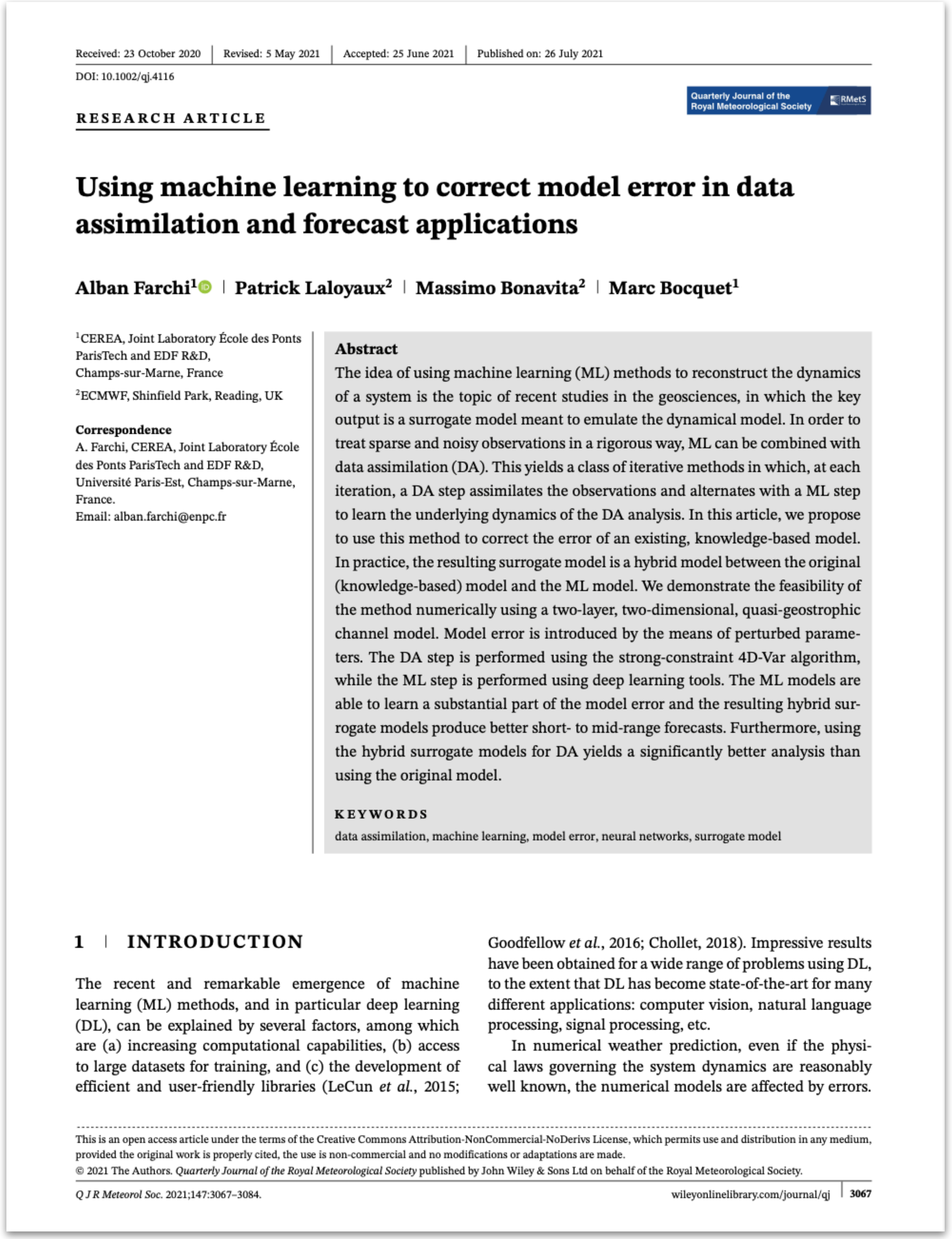
vs

Strong constraint **4DVar**

control : NN parameters







Received: 23 October 2020 | Revised: 5 May 2021 | Accepted: 25 June 2021 | Published on: 26 July 2021

DOI: 10.1002/qj.4116

Quarterly Journal of the Royal Meteorological Society RMetS

RESEARCH ARTICLE

Using machine learning to correct model error in data assimilation and forecast applications

Alban Farchi[1] | **Patrick Laloyaux[2]** | **Massimo Bonavita[2]** | **Marc Bocquet[1]**

[1]CEREA, Joint Laboratory École des Ponts ParisTech and EDF R&D, Champs-sur-Marne, France

[2]ECMWF, Shinfield Park, Reading, UK

Correspondence
A. Farchi, CEREA, Joint Laboratory École des Ponts ParisTech and EDF R&D, Université Paris-Est, Champs-sur-Marne, France.
Email: alban.farchi@enpc.fr

Abstract

The idea of using machine learning (ML) methods to reconstruct the dynamics of a system is the topic of recent studies in the geosciences, in which the key output is a surrogate model meant to emulate the dynamical model. In order to treat sparse and noisy observations in a rigorous way, ML can be combined with data assimilation (DA). This yields a class of iterative methods in which, at each iteration, a DA step assimilates the observations and alternates with a ML step to learn the underlying dynamics of the DA analysis. In this article, we propose to use this method to correct the error of an existing, knowledge-based model. In practice, the resulting surrogate model is a hybrid model between the original (knowledge-based) model and the ML model. We demonstrate the feasibility of the method numerically using a two-layer, two-dimensional, quasi-geostrophic channel model. Model error is introduced by the means of perturbed parameters. The DA step is performed using the strong-constraint 4D-Var algorithm, while the ML step is performed using deep learning tools. The ML models are able to learn a substantial part of the model error and the resulting hybrid surrogate models produce better short- to mid-range forecasts. Furthermore, using the hybrid surrogate models for DA yields a significantly better analysis than using the original model.

KEYWORDS

data assimilation, machine learning, model error, neural networks, surrogate model

1 | INTRODUCTION

The recent and remarkable emergence of machine learning (ML) methods, and in particular deep learning (DL), can be explained by several factors, among which are (a) increasing computational capabilities, (b) access to large datasets for training, and (c) the development of efficient and user-friendly libraries (LeCun *et al.*, 2015; Goodfellow *et al.*, 2016; Chollet, 2018). Impressive results have been obtained for a wide range of problems using DL, to the extent that DL has become state-of-the-art for many different applications: computer vision, natural language processing, signal processing, etc.

In numerical weather prediction, even if the physical laws governing the system dynamics are reasonably well known, the numerical models are affected by errors.

Learning models from noisy, incomplete data ?

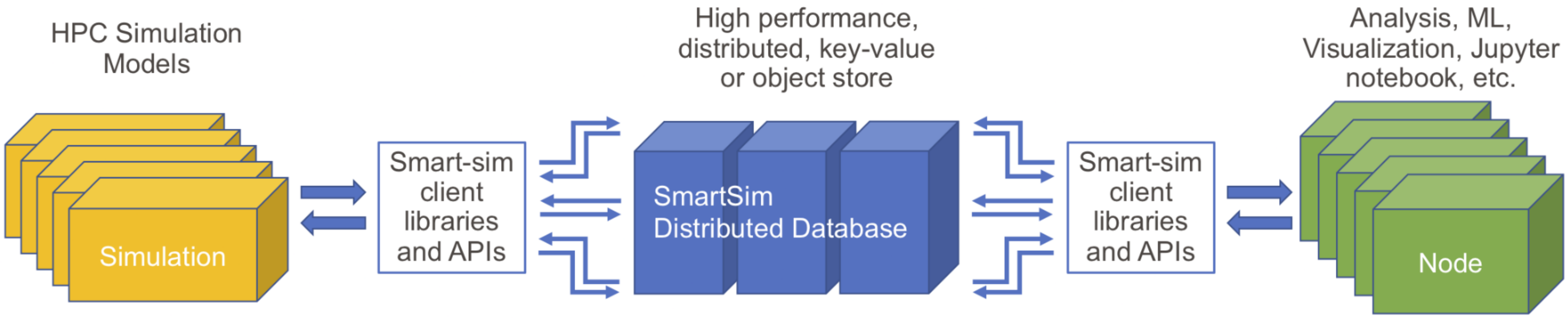
Summary

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

- Practical applications of end-to-end learning for subgrid closures in complex models may possibly leverage **differentiable emulators**
- End-to-end learning is equivalent to **strong-constraint 4DVar** and may therefore work with sparse and noisy target data (?)

ML-based closures in legacy models

NN-based closures, emulation



(Partee et al. 2021)

How to share NN-based model components ?

A posteriori learning of quasi-geostrophic turbulence parametrization: an experiment on integration steps

Hugo Frezat
Univ. Grenoble Alpes, CNRS UMR LEGI, Grenoble, France
Univ. Grenoble Alpes, CNRS UMR IGE, Grenoble, France
IMT Atlantique, CNRS UMR Lab-STICC, Brest, France
`hugo.frezat@univ-grenoble-alpes.fr`

Julien Le Sommer
Univ. Grenoble Alpes, CNRS UMR IGE, Grenoble, France

Ronan Fablet
IMT Atlantique, CNRS UMR Lab-STICC, Brest, France

Guillaume Balarac
Univ. Grenoble Alpes, CNRS UMR LEGI, Grenoble, France
Institut Universitaire de France (IUF), Paris, France

Redouane Lguensat
Learning, Data and Robotics Lab, ESIEA, Paris, France
LOCEAN-IPSL, Sorbonne Université, Institut Pierre Simon Laplace, Paris, France

Abstract

Modeling the subgrid-scale dynamics of reduced models is a long standing open problem that finds application in ocean, atmosphere and climate predictions where direct numerical simulation (DNS) is impossible. While neural networks (NNs) have already been applied to a range of three-dimensional flows with success, two dimensional flows are more challenging because of the backscatter of energy from small to large scales. We show that learning a model jointly with the dynamical solver and a meaningful *a posteriori*-based loss function lead to stable and realistic simulations when applied to quasi-geostrophic turbulence.

1 Introduction

Understanding and predicting the evolution of various natural systems would not be possible without simulations of turbulent flows. However, solving all the spatial and temporal scales of the associated partial differential equation (PDE), i.e., the Navier-Stokes equations remains computationally prohibitive. One popular solution (e.g. [14, 8, 4]) is to resolve only the largest scales and use subgrid closures (or physical parametrizations) to represent the smaller ones.

Recently, neural networks (NNs) have been proposed as a promising alternative to algebraic parametrizations in three-dimensional incompressible turbulence [6, 15, 5]. Two-dimensional problems, however, are more challenging due to the inverse cascade of energy that leads to negative viscosities, and it has been demonstrated that numerical stability of the trained model in decaying turbulence requires either the removal of negative eddy viscosities [13] or a large training dataset [9].

Fourth Workshop on Machine Learning and the Physical Sciences (NeurIPS 2021).

manuscript submitted to *Journal of Advances in Modeling Earth Systems (JAMES)*

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

Key Points:

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

Abstract

The use of machine learning for designing subgrid parameterizations for climate models is receiving growing attention. State-of-the-art strategies approach 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 models output coarse-grained in order to mimic coarse resolution models fields. By essence, these strategies optimize subgrid parameterizations to meet so-called *a priori* criteria. But the actual purpose of a subgrid parameterization is to obtain good performances 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 parameterizations obtained with different learning strategies at fixed computational complexity. We show that strategies based on *a priori* criteria yield parameterizations that tend to be unstable in direct simulations and describe how subgrid parameterizations can alternatively be trained end-to-end in order to meet *a posteriori* criteria. We illustrate that end-to-end learning strategies yield parameterizations that outperform known empirical and data-driven schemes in terms of performance, stability and ability to generalize to different flow configurations. These results suggest that there will be a need to adopt differentiable programming paradigms for climate models in the future.

Corresponding author: Hugo Frezat, `hugo.frezat@univ-grenoble-alpes.fr`

Frezat et al. 2021 : https://arxiv.org/abs/2111.06841

Frezat et al. (in prep for JAMES)