

Journal Pre-proof

Frame invariant neural network closures for Kraichnan turbulence

Suraj Pawar, Omer San, Adil Rasheed, Prakash Vedula

PII: S0378-4371(22)00885-8
DOI: <https://doi.org/10.1016/j.physa.2022.128327>
Reference: PHYSYA 128327

To appear in: *Physica A*



Please cite this article as: S. Pawar, O. San, A. Rasheed et al., Frame invariant neural network closures for Kraichnan turbulence, *Physica A* (2022), doi: <https://doi.org/10.1016/j.physa.2022.128327>.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2022 Elsevier B.V. All rights reserved.

Frame invariant neural network closures for Kraichnan turbulence

Suraj Pawar^a, Omer San^{a,*}, Adil Rasheed^b, Prakash Vedula^c^a*School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, Oklahoma 74078, USA*^b*Department of Engineering Cybernetics, Norwegian University of Science and Technology, N-7465, Trondheim, Norway.*^c*School of Aerospace & Mechanical Engineering, The University of Oklahoma, Norman, OK 73019, USA.*

Abstract

Numerical simulations of geophysical and atmospheric flows have to rely on parameterizations of subgrid scale processes due to their limited spatial resolution. Despite substantial progress in developing parameterization (or closure) models for subgrid scale (SGS) processes using physical insights and mathematical approximations, they remain imperfect and can lead to inaccurate predictions. In recent years, machine learning has been successful in extracting complex patterns from high-resolution spatio-temporal data, leading to improved parameterization models, and ultimately better coarse grid prediction. However, the inability to satisfy known physics and poor generalization hinders the application of these models for real-world problems. In this work, we put forth a frame invariant closure approach to improve the accuracy and generalizability of deep learning-based subgrid scale closure models by embedding physical symmetries directly into the structure of the neural network. Specifically, we utilized specialized layers within the convolutional neural network in such a way that desired constraints are theoretically guaranteed without the need for any regularization terms. We demonstrate our framework for a two-dimensional decaying turbulence test case mostly characterized by the forward enstrophy cascade. We show that our frame invariant SGS model (i) accurately predicts the subgrid scale source term, (ii) respects the physical symmetries such as translation, Galilean, and rotation invariance, and (iii) is numerically stable when implemented in coarse-grid simulation with generalization to different initial conditions and Reynolds number. This work opens up a possibility of connecting physics-based theories and data-driven modeling paradigms, and thus represents a promising step towards the development of physically consistent data-driven turbulence closure models.

Keywords: Equivariant neural networks, Subgrid scale closure modeling, Frame invariances, Geophysical flows

*Corresponding author.

Email addresses: supawar@okstate.edu (Suraj Pawar), osan@okstate.edu (Omer San), adil.rasheed@ntnu.no (Adil Rasheed), pvedula@ou.edu (Prakash Vedula)

1. Introduction

Computational modeling of turbulent flows remains a key issue in many engineering and geophysical applications. Turbulence involves a wide range of spatio-temporal scales that makes the direct numerical simulation (DNS) computationally infeasible for many complex systems. Coarse-graining approaches like large eddy simulation (LES) alleviate the computational burden by resolving only large and intermediate scales of the flow. The non-linearity of the Navier-Stokes equations introduces a subgrid scale (SGS) closure problem in LES which can be addressed via modeling of unresolved scales on the resolved flow quantities. The choice of the SGS model directly affects the accuracy of LES-based solution, and, therefore, the SGS modeling has been an active area of research for the past few decades [1, 2, 3, 4]. The development of SGS models has largely been driven by physical insights, mathematical considerations, and often problem-specific intuition. More recently, the availability of data from observations and high-resolution simulation along with advances in hardware and algorithms has fuelled interest in the development of data-driven turbulence models [5, 6, 7, 8].

The initial efforts towards data-driven SGS modeling include training a neural network to predict computationally expensive SGS model for channel flow [9] with the aim to speed-up LES. Similar frameworks includes applying neural network to determine the eddy-viscosity of the dynamic Smagorinsky model [10, 11], SGS model classification and blending [12], and reinforcement learning to predict SGS dissipation coefficient [13]. Deep learning (DL) has been applied to discovering new SGS models from the DNS data without any assumption of prior structural or functional form of the model [14, 15, 16, 17, 18, 19, 20]. The data-driven approach that employs convolutional neural network for learning the SGS model has also been used for different problems like two-dimensional decaying turbulence [11, 21, 22], three-dimensional decaying homogeneous isotropic turbulence [23], momentum forcing in ocean models [24], and subgrid-scale scalar flux modeling [25]. Moreover, neural networks have also been utilized to learn the optimal map between filtered and unfiltered variables in the approximate deconvolution framework for SGS modeling [26, 27]. Apart from SGS closure modeling, machine learning (ML) and in particular DL is being increasingly applied for different problems in fluid mechanics, like superresolution of turbulent flows [28, 29], Reynolds-Average Navier-Stokes (RANS) closure modeling [30, 31, 32], data assimilation [33, 34, 35, 36], spatio-temporal forecasting of fluid flows [37, 38, 39] and reduced-order modeling [40, 41, 42, 43, 44].

Despite their early success, ML models are faced with an array of challenges, such as poor generalization, lack of interpretability, and in some cases, violation of the known governing laws of the physical systems. For example, the SGS model derived through supervised learning may be numerically unstable, and diverge from the original trajectory, and this issue is exposed in many studies [15, 45, 23, 46]. These issues can be addressed by leveraging our prior knowledge about the physical systems into an ML model. Readers are referred to recent review articles on physics-informed machine learning [47, 48] that detail different methods of incorporating physics into ML models and discuss the capabilities and limitations of these methods. In the context of SGS modeling, there are many ways to embed physical constraints into the ML model. One such method for constructing a robust and generalizable SGS

model is through the selection of suitable non-dimensionalized input and output quantities of the ML model to ensure that the known symmetries are respected [49]. Another class of methods pertains to the customized neural network architectures that encode the prior physical or mathematical knowledge as hard constraints. Some of the examples of this methods applied in fluid dynamics are tensor basis neural network [50], transformation invariant neural network [25], physics-embedded neural network [51], spatial transformer [52, 53], and equivariant networks [54, 55, 56].

In this work, we address the challenges associated with data-driven turbulence modeling by introducing a frame invariant convolutional neural network (FI-CNN) for SGS closure model discovery. Specifically, we select model inputs that are Galilean invariant, and replace the convolution operation with group convolutions [57, 58] to embed rotation invariance. Therefore, the FI-CNN preserves various symmetries, including translation, Galilean, and rotation both during training and inference. This makes the FI-CNN framework physically consistent and robust to extrapolation, and consequently, it produces accurate and stable results in their *a posteriori* deployment. We demonstrate our framework for two-dimensional turbulence which is often used as a prototypical test case for large-scale geophysical flows [59, 60, 61, 62, 63]. Although we focus on SGS closure model development in this study, this framework has a promising application for many scientific problems where physical symmetries are very common. For example, there are several invariant finite-difference schemes based on equivariant moving frames that preserve Lie symmetries that have been developed for the solution of partial differential equations (PDEs) via consideration of modified forms of the underlying PDEs [64, 65, 66]. These symmetries can be exploited along with data-driven discretization [67] to design numerical schemes that are more accurate than their non-invariant counterpart.

This paper is organized as follows. In Sec 2, the symmetries of Navier-Stokes equations and the SGS closure modeling problem for two-dimensional turbulence is introduced. The detailed procedure on how to embed frame symmetries, including translation, Galilean, and rotation invariance within the CNN is provided in Sec 3. In Sec 4, the details on data generation and training are discussed. The performance of the FI-CNN in the *a priori* and *a posteriori* settings along with a detailed discussion of the results are presented in Sec 5. Finally, the concluding remarks and summary of the work are given in Sec 6.

2. SGS Closure Modeling

2.1. Symmetries of Navier-Stokes Equations

The Navier-Stokes equations governing incompressible fluid flows can be written in primitive variable (velocity–pressure) form as

$$\nabla \cdot \mathbf{u} = 0, \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u}, \quad (2)$$

where \mathbf{u} is the velocity, p is the pressure, ρ is the density, and ν is the kinematic viscosity of the fluid. The governing equations for LES (also called as the filtered Navier-Stokes

equations) are obtained by applying a spatial filter operation and it can be written as follows

$$\nabla \cdot \bar{\mathbf{u}} = 0, \quad (3)$$

$$\frac{\partial \bar{\mathbf{u}}}{\partial t} + \bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}} = -\frac{1}{\rho} \nabla \bar{p} + \nu \nabla^2 \bar{\mathbf{u}} + \nabla \cdot \underbrace{(\bar{\mathbf{u}}\bar{\mathbf{u}} - \bar{\mathbf{u}}\bar{\mathbf{u}})}_{\boldsymbol{\tau}_{\text{SGS}}}, \quad (4)$$

80 where the overbar is used to denote the filtered variables and $\boldsymbol{\tau}_{\text{SGS}}$ is subgrid-scale stress tensor. The problem of determining subgrid-scale stress tensor $\boldsymbol{\tau}_{\text{SGS}}$ using the filtered variables is called the subgrid scale closure problem in LES.

There are many possible SGS closure models and any mathematical, physical constraints will lead to a specific type of SGS model. Requirements such as frame invariance, realizability, finite kinetic energy can act as guiding principles for a satisfactory SGS closure model, and readers are referred to [3] for more details. The frame invariance constraint on the SGS model is derived by enforcing the symmetry of the original Navier-Stokes equations [2, 1, 68, 3] upon the filtered Navier-Stokes equations with the SGS closure model. Let G denote a group of transformation acting on space-time functions $\mathbf{u}(\mathbf{x}, t)$. We say that the group G is a symmetry group of the Navier-Stokes equations if, for all \mathbf{u} which are solutions of the Navier-Stokes equations, and all $g \in G$, the function $g\mathbf{u}$ is also a solution [69]. The frame invariance constraint involves preservation of the symmetry property of the original Navier-Stokes equations to translation, Galilean, and rotation transformations, and they can be written as follows

- 95 • Space-translation: $g_{\boldsymbol{\delta}}^{\text{space}} \mathbf{u}(\mathbf{x}, t) = \mathbf{u}(\mathbf{x} - \boldsymbol{\delta}, t)$, $\forall \boldsymbol{\delta} \in \mathbb{R}^3$, where $g_{\boldsymbol{\delta}}^{\text{space}}$ is the space-translation operator with the arbitrary displacement $\boldsymbol{\delta}$.
- Galilean transformation: $g_{\boldsymbol{\alpha}}^{\text{Gal}} \mathbf{u}(\mathbf{x}, t) = \mathbf{u}(\mathbf{x} - \boldsymbol{\alpha}t, t) + \boldsymbol{\alpha}$, $\forall \boldsymbol{\alpha} \in \mathbb{R}^3$, where $g_{\boldsymbol{\alpha}}^{\text{Gal}}$ is the Galilean operator and $\boldsymbol{\alpha}$ is a fixed but arbitrary constant vector.
- 100 • Space-rotations: $g_{\mathbf{A}}^{\text{rot}} \mathbf{u}(\mathbf{x}, t) = \mathbf{A}\mathbf{u}(\mathbf{A}^{-1}\mathbf{x}, t)$, where $g_{\mathbf{A}}^{\text{rot}}$ is the rotation operation and $\mathbf{A} \in SO(3)$.

Imposing the symmetry preservation constraint give some structure to the the SGS model, and this insights have been extensively used in turbulence models [70, 2, 3]. We make use of these symmetries as physical constraints while building a frame invariant data-driven SGS model.

105 2.2. Two-dimensional Turbulence

In this work, we are interested in the SGS modeling for two-dimensional turbulence that is usually applied for modeling geophysical flows in the atmosphere and ocean [60, 59] where rotation and stratification dominate, and the most efficient way to model it is using the vorticity transport equation. Taking the curl of Eq. 2 yields the Navier-Stokes equations

in vorticity-velocity formulation, and, for incompressible fluid flows, it can be written as follows

$$\frac{\partial \omega}{\partial t} + (\mathbf{u} \cdot \nabla) \omega = \nu \nabla^2 \omega, \quad (5)$$

where ω is the vorticity, and for two-dimensional flows, we have $\omega = \partial v / \partial x - \partial u / \partial y$. A scalar function called the streamfunction is defined in such a way that the continuity equation is satisfied if the velocity expressed in terms of the streamfunction is substituted in the continuity equation. This leads to the definition of velocity in terms of the streamfunction as follows

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x}, \quad (6)$$

where ψ is the streamfunction. The Poisson equation relating the vorticity and streamfunction is obtained by substituting the above velocity components in the definition of vorticity. Thus, we have

$$\nabla^2 \psi = -\omega. \quad (7)$$

It is convenient to write Eq. 5 in the vorticity-streamfunction formulation as follows

$$\frac{\partial \omega}{\partial t} + J(\omega, \psi) = \frac{1}{\text{Re}} \nabla^2 \omega, \quad (8)$$

$$J(\omega, \psi) = \frac{\partial \omega}{\partial x} \frac{\partial \psi}{\partial y} - \frac{\partial \omega}{\partial y} \frac{\partial \psi}{\partial x}, \quad (9)$$

where $J(\cdot, \cdot)$ is the Jacobian (or the nonlinear term), and Re is the Reynolds number of the flow. The above equation is also called the vorticity transport equation.

The filtered Navier-Stokes equations for two-dimensional turbulence is obtained by applying a spatial filtering operation to Eq 8 as follows

$$\frac{\partial \bar{\omega}}{\partial t} + \overline{J(\omega, \psi)} = \frac{1}{\text{Re}} \nabla^2 \bar{\omega}. \quad (10)$$

The above equation can be rewritten as

$$\frac{\partial \bar{\omega}}{\partial t} + J(\bar{\omega}, \bar{\psi}) = \frac{1}{\text{Re}} \nabla^2 \bar{\omega} + \Pi, \quad (11)$$

where the overbar quantities represent filtered variables and are evolved on a grid that is significantly coarse compared to the DNS resolution. The effect of the unresolved scales due to truncation of high wavenumber flow scales is encapsulated in a subgrid scale (SGS) source term Π and must be modeled solely based on the resolved variables $(\bar{\omega}, \bar{\psi})$. Mathematically, the true SGS source term Π can be expressed as

$$\Pi = J(\bar{\omega}, \bar{\psi}) - \overline{J(\omega, \psi)}. \quad (12)$$

The functional and structural models are the most commonly used approaches for modeling the SGS closure term in LES of turbulent flows [2]. The functional models are based

110 on the concept of eddy viscosity where the effect of unresolved scales are approximated by artificial dissipation [71, 72]. The functional models can be further improved by dynamic adaptations of the coefficients that control the dissipation of the model and are determined adaptively by the use of a low-pass spatial test filter [73, 74, 75]. Although the dynamic formulation allows for spatial and temporal variation of coefficients in the eddy viscosity model, the ensemble averaging procedure does not allow for true back-scattering in order to limit the growth of numerical instabilities during the *a posteriori* testing [76, 77]. The structural models on the other hand aim at obtaining an accurate approximation of the SGS term and are based on the approximate deconvolution procedure [78, 79] and scale-similarity arguments [80]. Scale-similarity models address the SGS closure term by extrapolation from the smallest resolved scales to unresolved scales and have found to be the most accurate in *a priori* testing [2, 81]. However, numerical instabilities have been reported with scale-similarity models, and this has led to development of many mixed models with additional eddy viscosity term for stability reasons [82, 83, 84]. More recently, data-driven methods are emerging as a new paradigm to build turbulence closure models by extracting information from the data, and are seen as the potential applications to address the limitations of existing SGS models [5, 6, 85].

3. Frame invariant SGS closure model

In this work, we consider the frame invariance property that must be satisfied by any SGS model and demonstrate how to include them within a neural network as hard constraints. The SGS source term Π is approximated using a neural network as shown below

$$\tilde{\Pi} \approx \mathcal{M}(\bar{\omega}, \bar{\psi}), \quad (13)$$

130 where \mathcal{M} is a neural network-based model, and $\tilde{\Pi}$ is the approximation of true SGS source term Π . We remark here that the vorticity is defined using the spatial derivative of the velocity field, and, therefore it is invariant to Galilean transformations. Additionally, the streamfunction is computed using the vorticity, and therefore both the inputs to our model are Galilean invariant. We now discuss how to embed the translation and rotational invariance/symmetry properties into the neural network-based model.

3.1. Translation invariance

In this work, we use the convolutional neural network (CNN) for learning the SGS closure model based on filtered vorticity and streamfunction as the model inputs. The CNN is an attractive choice for high-dimensional data and it does not suffer from the curse of dimensionality due to its weight-sharing feature. The CNN is composed of many convolutional layers and each of the layers is parameterized by filters, also called kernels, that has to be learned through training. Let $\underline{f}, \underline{k} : \mathbb{R}^2 \rightarrow \mathbb{R}^{N_c}$ be vector-valued two-dimensional features and kernel, i.e., $\underline{f} = (f_1, \dots, f_{N_c})$ and $\underline{k} = (k_1, \dots, k_{N_c})$, then the convolutional operation can be defined as follows

$$(\underline{k} \star \underline{f})(\mathbf{x}) = \sum_{c=1}^{N_c} \int_{\mathbb{R}^2} k_c(\mathbf{x} - \mathbf{x}') f_c(\mathbf{x}') d\mathbf{x}', \quad (14)$$

where \mathbf{x}' is a dummy variable spanning over \mathbb{R}^2 space. The convolutional layer maps a feature vector $\underline{f}^{(l-1)} : \mathbb{R}^2 \rightarrow \mathbb{R}^{N_{l-1}}$ with N_{l-1} channels to feature vector $\underline{f}^{(l)} : \mathbb{R}^2 \rightarrow \mathbb{R}^{N_l}$ using a set of N_l kernels $\mathbf{k}^{(l)} := (k_1^{(l)}, \dots, k_{N_l}^{(l)})$ and this operation can be defined as

$$\underline{f}^{(l)} = \zeta(\mathbf{k}^{(l)} \star \underline{f}^{(l-1)}) := \zeta(k_1^{(l)} \star \underline{f}^{(l-1)}, \dots, k_{N_l}^{(l)} \star \underline{f}^{(l-1)}), \quad (15)$$

where ζ is an activation function. The parameters of the kernel are shared for the whole image as the kernel is convolved relative to the position about \mathbf{x} and this aspect of the relative motion makes the CNN translation invariant. Although we present the convolution operation with continuous kernels, convolutional layers are equipped with discretized-filtering operations in their practical implementation. From here on, we refer to the model built using convolutional layers and nonlinear activation function as \mathcal{M}_{CNN} . The inputs to our model are the vorticity and streamfunction and the output is the SGS source term. Therefore, the learning map for \mathcal{M}_{CNN} can be expressed as

$$\mathcal{M}_{\text{CNN}} : \{\bar{\omega}, \bar{\psi}\} \in \mathbb{R}^2 \rightarrow \mathbb{R}^2 \mapsto \{\tilde{\Pi}\} \in \mathbb{R}^2 \rightarrow \mathbb{R}^1, \quad (16)$$

135 where $\tilde{\Pi}$ is the predicted SGS source term from a neural network.

3.2. Rotation invariance

The rotational invariance of the SGS model requires that it maps as a tensor under the coordinate rotation [2]. As discussed in Section 3.1, the CNN is often invariant to only translation and not for other groups of transformations. However, there are recent developments
 140 on this front to exploit polar mapping of input images to convert rotation to translation [86]. In this work, we apply the group equivariant convolutions within the $E(2)$ -CNN framework [87] for embedding rotational symmetry. The first roto-translation equivariant CNN was called the group convolutional neural network (GCNN) and it considered the rotations by multiples of $\pi/2$ [57]. The GCNN was further augmented by defining filters in terms of the
 145 steerable basis that are equivariant to rotations by multiples of $2\pi/N$, with $N > 4$ [88]. The $E(2)$ -CNN library is based on the framework of steerable CNNs [89, 58] and it has different options for the group that takes the form of the semi-direct group $H = \mathbb{R}^2 \rtimes G$ where the group $G \leq O(2)$ (here $O(2)$ is the group of continuous rotations and reflections). For example, the group $H = \mathbb{R}^2 \rtimes SO(2) = SE(2)$ is the semi-direct product of the group of planar
 150 translations \mathbb{R}^2 and continuous rotations $SO(2)$. In this work, we utilize the cyclic group $G = C_N$ containing the discrete rotations of $2\pi/N$ (i.e., $H = \mathbb{R}^2 \rtimes C_N$). For a large value of N , the difference between continuous rotations and discrete rotations is indistinguishable due to space discretization.

A full understanding of the steerable CNNs requires some knowledge of the group representation theory, but the implementation of the steerable CNNs is similar to ordinary CNNs. Readers are suggested to read Weiler and Cesa [87] and references therein for a more comprehensive discussion on the general framework of steerable CNNs. Here, we briefly explain the G -equivariant convolutions. A G -convolution between a vector-valued two-dimensional

image $\underline{f} : \mathbb{R}^2 \rightarrow \mathbb{R}^{N_c}$ and a filter $\underline{k} : \mathbb{R}^2 \rightarrow \mathbb{R}^{N_c}$ where $\underline{f} = (f_1, \dots, f_{N_c})$ and $\underline{k} = (k_1, \dots, k_{N_c})$ can be expressed as follows

$$(\underline{k} \tilde{\star} \underline{f})(g) = \sum_{c=1}^{N_c} \int_{\mathbb{R}^2} k_c(g^{-1} \mathbf{x}') f_c(\mathbf{x}') d\mathbf{x}', \quad (17)$$

where $g = (\mathbf{x}, \theta) \in H = \mathbb{R}^2 \rtimes C_N$, $\mathbf{x}' \in \mathbb{R}^2$, and $\tilde{\star}$ denotes the group correlation operation under joint translation and rotation. This operation corresponds to lifting of the data on two-dimensional space to the data that lives on a three-dimensional position orientation space H . The first layer maps a two-dimensional image $\underline{f}^{(l-1)} : \mathbb{R}^2 \rightarrow \mathbb{R}^{N_{l-1}}$ with N_{l-1} channels at $(l-1)$ th layer to H vector image $\underline{F}^{(l)} : H \rightarrow \mathbb{R}^{N_l}$ using a set of N_l kernels $\mathbf{k}^{(l)} := (k_1^{(l)}, \dots, k_{N_l}^{(l)})$ as follows

$$\underline{F}^{(l)} = \zeta(\mathbf{k}^{(l)} \tilde{\star} \underline{f}^{(l-1)}) := \zeta(k_1^{(l)} \tilde{\star} \underline{f}^{(l-1)}, \dots, k_{N_l}^{(l)} \tilde{\star} \underline{f}^{(l-1)}). \quad (18)$$

Since the \underline{F} is a function on H , the filters from the second layer onward should also be functions on H . The subsequent group convolutions are defined as [88, 90]

$$(\underline{K} \tilde{\star} \underline{F})(g) = \sum_{c=1}^{N_c} \int_H K_c(g^{-1} h) F_c(h) dh. \quad (19)$$

A group convolution layer is defined by a set of H kernels $\mathbf{K} := (K_1^{(l)}, \dots, K_{N_l}^{(l)})$ that maps $\underline{F}^{(l-1)}$ with N_{l-1} channels to $\underline{F}^{(l)}$ with N_l channels as shown below

$$\underline{F}^{(l)} = \zeta(\mathbf{K}^{(l)} \tilde{\star} \underline{F}^{(l-1)}) := \zeta(K_1^{(l)} \tilde{\star} \underline{F}^{(l-1)}, \dots, K_{N_l}^{(l)} \tilde{\star} \underline{F}^{(l-1)}). \quad (20)$$

155 Finally, the feature field at the last layer can be synthesized from H space to \mathbb{R}^2 space. The user interface of the $E(2)$ -CNN library [87] hides most of the intricacies of group theory, solutions of the steerable kernels space constraints, and requires users to specify only the transformation laws of the feature spaces. We use the regular representation for all hidden layers and the action of regular representation is given by permutation matrices (Appendix B in [87]).

To summarize, we embed the Galilean invariance into our neural network model by using input features (i.e., vorticity and streamfunction) that are invariant to Galilean transformation. The translation variance is automatically embedded through the use of a convolutional neural network to learn the mapping from input features to the subgrid-scale closure term. Lastly, the rotation variance is embedded as a hard constraint by using group convolution layers that are invariant to discrete rotations of $2\pi/N$. From here on, the model built using the equivariant CNN is called $\mathcal{M}_{\text{FI-CNN}}$. The learning map for $\mathcal{M}_{\text{FI-CNN}}$ is the same as the \mathcal{M}_{CNN} and can be written as follows

$$\mathcal{M}_{\text{FI-CNN}} : \{\bar{\omega}, \bar{\psi}\} \in \mathbb{R}^2 \rightarrow \mathbb{R}^2 \mapsto \{\tilde{\Pi}\} \in \mathbb{R}^2 \rightarrow \mathbb{R}^1, \quad (21)$$

160 where $\tilde{\Pi}$ is the predicted SGS source term from a neural network.

4. Data Generation and Training

The parameters of the neural network based SGS models are learned through supervised training that requires a set of labeled inputs and outputs, usually obtained from direct numerical simulation (DNS). The dataset should encompass a range of dynamics that is expected to be reproduced by the SGS model. The data for training is generated from DNS of two-dimensional Kraichnan turbulence in a doubly periodic square domain with $L_x \times L_y = [0, 2\pi] \times [0, 2\pi]$, and the domain is discretized using 2048^2 degrees of freedom. Our DNS solver is based on a second-order accurate energy-conserving Arakawa scheme [91] for the nonlinear Jacobian and second-order accurate finite-difference scheme for the Laplacian of the vorticity. The elliptic equation for the relationship between the streamfunction and vorticity is solved using a second-order accurate FFT-based Poisson solver, and the time integration is performed with a third-order accurate Runge-Kutta method. The vorticity distribution at the start of the simulation is initialized based on the energy spectrum given by [92]

$$E(k) = Ak^4 \exp\left(-\left(\frac{k}{k_p}\right)^2\right), \quad (22)$$

where $A = 4k_p^{-5}/3\pi$ and $k = |\mathbf{k}| = \sqrt{k_x^2 + k_y^2}$. For our numerical experiments, we use $k_p = 10$. The initial vorticity distribution in Fourier space is obtained through the introduction of random phase as follows

$$\tilde{\omega}(\mathbf{k}) = \sqrt{\frac{k}{\pi} E(k)} e^{i\xi(\mathbf{k})}, \quad (23)$$

where the phase function is given by $\xi(\mathbf{k}) = \phi(\mathbf{k}) + \eta(\mathbf{k})$. Here, $\phi(\mathbf{k})$ and $\eta(\mathbf{k})$ are independent random values chosen in $[0, 2\pi]$ at each grid point in the first quadrant of the $k_x - k_y$ plane (i.e., $k_x, k_y \geq 0$). The phase function for other quadrants is obtained through conjugate relations as follows

$$\xi(\mathbf{k}) = -\phi(\mathbf{k}) + \eta(\mathbf{k}) \text{ for } k_x < 0 \text{ and } k_y \geq 0, \quad (24)$$

$$\xi(\mathbf{k}) = -\phi(\mathbf{k}) - \eta(\mathbf{k}) \text{ for } k_x < 0 \text{ and } k_y < 0, \quad (25)$$

$$\xi(\mathbf{k}) = \phi(\mathbf{k}) - \eta(\mathbf{k}) \text{ for } k_x \geq 0 \text{ and } k_y < 0, \quad (26)$$

$$(27)$$

Further details on the problem setup and the numerical schemes can be found in our previous work [84]. Different realizations of the initial vorticity field can be obtained by using different phase functions with a different seed for random value generation.

165 The DNS is performed from time $t = 0$ to $t = 4$ with the time step $\Delta t = 5 \times 10^{-4}$. In the Kraichnan turbulence problem, the initial vorticity field is dominated by a population of vortices and small-scale structure starts appearing as the flow evolves. The initial spin-up time from $t = 0$ to 0.5 is neglected and we start collecting the data for training from time $t = 0.5$. From time $t \approx 0.5$, the flow has started following Kraichnan–Batchelor–Leith
170 (KBL) theory [93, 94, 72] of energy cascade where energy is transferred from the smaller scales to the larger scales. From time $t \approx 0.5$ onward, large coherent vortices start emerging

through vortex merging mechanism and viscous dissipation of small-scale structures. The vorticity field and angle-averaged energy spectrum are displayed in Fig. 1 and we can see that the energy spectrum has started exhibiting k^{-3} scaling from approximately $t = 0.5$.

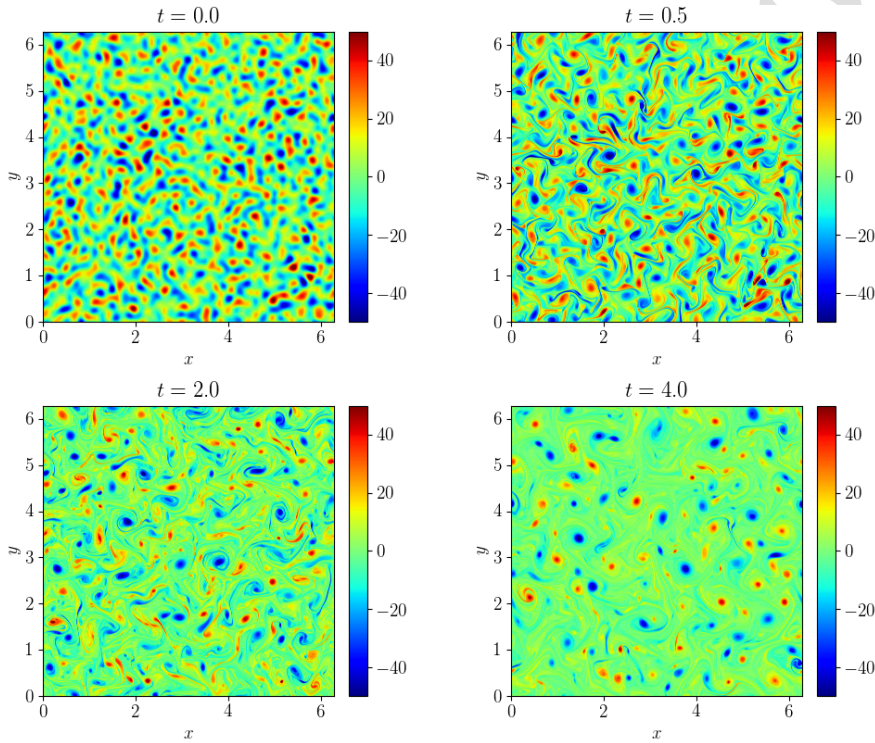


Figure 1: Visualization of the vorticity field and energy spectrum at different time instances for $\text{Re} = 16000$ with grid resolution 2048×2048 .

175 The data for training a neural network-based SGS model is stored at every $20\Delta t$, i.e., we have 350 snapshots of the vorticity and streamfunction between $t = 0.5$ to 4.0. We emphasize here that the neural network-based SGS model is trained only for a single Reynolds number $\text{Re} = 16000$ and we assess the performance of the model for Reynolds number up to $\text{Re} = 128000$. The filtered DNS data for training is obtained by first applying a Gaussian filter transfer function to the DNS data and then coarse-graining the filtered solution to the LES grid [95, 22]. The Gaussian filter provides a smooth transition between resolved and subgrid scales and is also positive definite in physical and wave space [96, 97]. Additionally, our numerical solver is in physical space, and therefore we select the Gaussian filter instead of a spectral cut-off filter. The coarse-grid level for LES is 256^2 which corresponds to 64 times
180 fewer spatial degrees of freedom compared to DNS.

We do not pre-process the filtered DNS data before training as the DNS data is generated

from a non-dimensionalized vorticity transport equation. The total data is divided into 80% of the data for training and 20% for the validation set. While the input and output of both \mathcal{M}_{CNN} and $\mathcal{M}_{\text{FI-CNN}}$ are the same, the user needs to specify the type of representation for intermediate feature field while constructing an FI-CNN [87], similar to the number of kernels for CNN. We use the kernel size of 5×5 , six hidden layers and ReLU activation function for the \mathcal{M}_{CNN} and $\mathcal{M}_{\text{FI-CNN}}$. The number of kernels for the CNN and FI-CNN models is set to 30 and 16, respectively. With these hyperparameters, the number of trainable parameters is roughly the same around $O(1.1 \times 10^5)$ for both models. Both the models are trained for 100 iterations using an Adam optimizer. Fig. 2 shows the history of training loss versus iterations for both neural network-based SGS models and we can observe that the loss for $\mathcal{M}_{\text{FI-CNN}}$ is almost one order magnitude less than the loss for \mathcal{M}_{CNN} . This can be attributed to rotational invariances incorporated in the $\mathcal{M}_{\text{FI-CNN}}$ against \mathcal{M}_{CNN} , which is only invariant to translation and Galilean transformation. For both neural network-based SGS models, we use the parameters (i.e., weights) corresponding to minimum validation loss obtained while training the neural network.

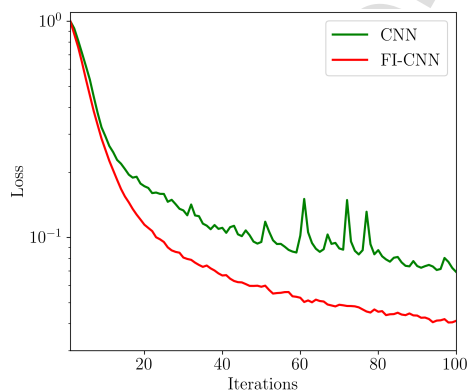


Figure 2: History of the training loss versus iterations for \mathcal{M}_{CNN} and $\mathcal{M}_{\text{FI-CNN}}$.

5. Numerical Results

In this section, we first outline the numerical results of our framework in the *a priori* settings where the neural network-based models are utilized in predicting the SGS source term. We analyze the capability of \mathcal{M}_{CNN} and $\mathcal{M}_{\text{FI-CNN}}$ in incorporating the frame-invariance property over the testing data. Then, we present the results of *a posteriori* LES coupled with neural network-based SGS models and evaluate their performance using numerous statistical metrics.

Before we present our results for the Kraichnan turbulence problem, we illustrate the capability of CNN and FI-CNN models to satisfy rotational equivariance using a simple example of a vortex merging test case. In this test case, two vortices of the same sign are

separated by a certain distance from each other along a parallel axis and the merging of vortices occurs ending as a single, nearly axisymmetric, final vortex [98]. The CNN and FI-CNN models are trained using the data generated for horizontally aligned vortices initial condition, i.e., the vorticity field shown in the top row and first column in Figure 3. During the inference, both CNN and FI-CNN models are able to predict the SGS source term with a sufficient level of accuracy. However, if we test the CNN model for vertically aligned vortices (bottom row and first column in Figure 3), we observe that the predicted SGS source term is inaccurate and the rotational equivariance is not satisfied. However, the FI-CNN model is successful in preserving the rotational symmetry and correctly captures the 90° rotation of the SGS source term. Therefore, by designing a model that respects certain physical symmetries, we can guarantee that the network is robust and generalizes well to different scenarios.

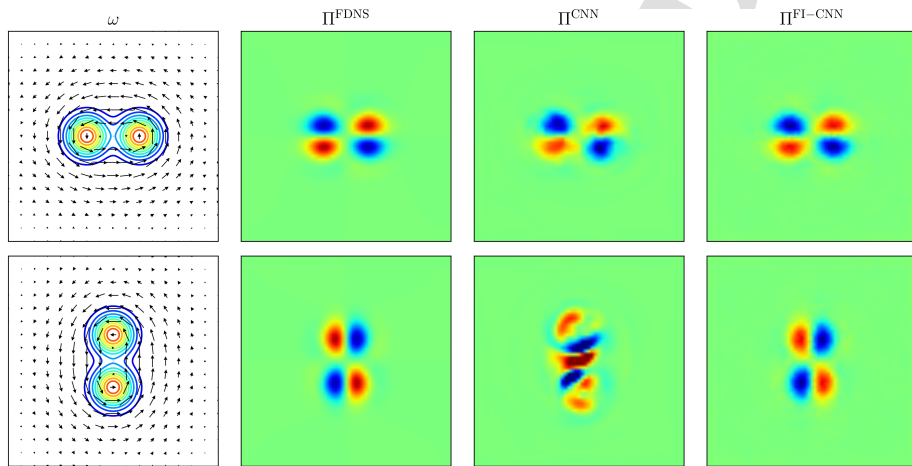


Figure 3: Illustration of rotational symmetry for the initial vorticity field of two Gaussian-distributed vortices. The CNN and FI-CNN models are trained using the data generated for horizontally aligned vortices, i.e., the top row and first column. When data-driven models are used for vertically aligned vortices, the CNN model fails to capture the rotational equivariance (the bottom row and third column), while the FI-CNN model correctly captures the rotational equivariance (the bottom row and fourth column).

5.1. A Priori investigation

Here, we assess the performance of neural network-based models in predicting the SGS source term compared to the true SGS source term for the out-of-training data. The out-of-training data is obtained for a different initial condition and corresponds to 70 snapshots stored randomly between time $t = 0.5$ to $t = 4.0$. We remark here that the initial energy spectrum for the testing data is also given by Eq. 22 and the difference is due to a different phase function. Fig. 4 displays the probability distribution function and cumulative distribution function for the test data. There is a very good agreement between the true SGS

source term and the predicted SGS source term from both models. However, we notice that the $\mathcal{M}_{\text{FL-CNN}}$ is more accurate near the tails of the distribution (Fig 4, left) compared to \mathcal{M}_{CNN} . This difference is also observable in the cumulative distribution function of true and predicted SGS source terms and is highlighted in the zoom-in portion (Fig 4, right). Based on these results, we may conclude that both neural network-based SGS model has learned the relationship between filtered quantities and the SGS source term. Both models are able to produce viable physical results for the completely unseen data with similar physics.

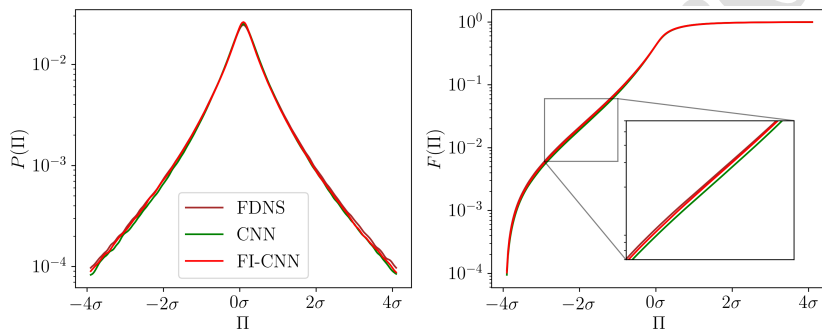


Figure 4: Probability distribution function (left) and cumulative distribution function (right) of the SGS source term over the entire testing dataset. The testing dataset corresponds to 70 snapshots selected randomly between time $t = 0.5$ to $t = 4.0$ for the initial condition different from the one used in training and σ is the standard deviation of the data.

Next, we evaluate neural network-based models in respecting rotational symmetry on the test data. Specifically, we perturbed the test data based on the rotation transformation, and generate multiple test datasets. Then, we compute the root mean squared error (RMSE) for each dataset, and calculate the expected value and variance for all test datasets. The rotation matrix \mathbf{A} is sampled uniformly between $[0^\circ, 360^\circ]$ in the multiple of 90° . If the rotation symmetry constraint is satisfied strictly, then the RMSE will be the same for each test dataset leading to zero variance for the RMSE metric. The results in Table 1 demonstrate the remarkable ability of $\mathcal{M}_{\text{FL-CNN}}$ to respect the rotation symmetry in contrast to \mathcal{M}_{CNN} which violates this symmetry. Furthermore, the expected value of RMSE for $\mathcal{M}_{\text{FL-CNN}}$ is one order of magnitude lower than \mathcal{M}_{CNN} and is consistent with the training loss (Fig. 2). The performance of both models is substantially accurate in terms of the Pearson's cross-correlation coefficient, with $\mathcal{M}_{\text{FL-CNN}}$ slightly better than \mathcal{M}_{CNN} . We note here that it is relatively straightforward to embed Galilean invariance constraint within neural network-based SGS model through intelligent selection of model inputs, and translation invariance through simple CNN. However, incorporating rotational symmetry in a neural network-based SGS model is more complex and requires special consideration. Although a relatively simple method like data augmentation can be utilized to impose the rotation symmetry as a soft constraint, it does not satisfy rotation invariance strictly [25]. The strict enforcement of rotation symmetry is challenging and requires the use of tailored neural network architecture,

Table 1: Evaluation of the rotational symmetry constraints provided by \mathcal{M}_{CNN} and $\mathcal{M}_{\text{FL-CNN}}$. The expected value and variance of the root mean squared error on the SGS source term predicted by both models is computed from many realizations (20 ensembles) on the testing data. The testing dataset corresponds to 70 snapshots selected randomly for the initial condition different from the one used for training. The rotational angle \mathbf{A} is sampled uniformly between $[0^\circ, 360^\circ]$ in the multiple of 90° and is used in the rotational operator $g_{\mathbf{A}}^{\text{rot}}$. The Pearson’s cross-correlation coefficient between the predicted SGS source term and the filtered DNS solution is computed as $\mathcal{P}(X, Y) = \text{cov}(X, Y) / \sigma_X \sigma_Y$.

Metric	\mathcal{M}_{CNN}	$\mathcal{M}_{\text{FL-CNN}}$
$E[\mathcal{L}]$	10.6941	7.3462
$\sigma[\mathcal{L}]$	4.2442×10^{-2}	5.6587×10^{-8}
$\mathcal{P}(X, Y)$	0.9600	0.9776

such as equivariant CNN.

5.2. *A Posteriori deployment*

260 We now evaluate the performance of neural network-based SGS models in the LES of Kraichnan turbulence. The spatial resolution for LES is reduced by a factor of eight in each direction and this gives us 256^2 degrees of freedom. The time step for LES simulation is ten times larger compared to the DNS, i.e., $\Delta t_{\text{LES}} = 5 \times 10^{-3}$. The performance of neural network-based SGS models is compared with the widely used dynamic Smagorinsky model (DSM) [73, 74]. The *a posteriori* deployment is a rigorous task for any data-driven SGS model due to the presence of numerical instabilities, and the challenges and remedies have been highlighted in many studies [15, 12, 99, 22, 23, 100, 101]. For example, Maulik et al. [15] and Zhou et al. [101] achieved the stable LES results by truncating SGS source term corresponding to negative eddy viscosity. Stoffer et al. [100] attained stable *a posteriori* results by resorting to artificially introducing additional dissipation (via eddy-viscosity models). Guan et al. [22] provided sufficient amount of data during training to obtain a stable *a posteriori* results. While the exact reason for this behavior is unknown, several issues such as error accumulation, aliasing errors, numerical instability, extrapolation beyond the training data, chaotic nature of turbulence, presence of multiple attractors might be responsible for unstable *a posteriori* simulation [23, 100, 6, 46].

275 From our *a posteriori* simulation, it is revealed that \mathcal{M}_{CNN} is unstable, while $\mathcal{M}_{\text{FL-CNN}}$ is able to produce a stable and physical solution without any kind of clipping or by adding artificial dissipation. We note here that, perhaps \mathcal{M}_{CNN} can also achieve stable *a posteriori* simulation, provided there is sufficient data available for training or some kind of post-processing is carried out for the predicted SGS source term. However, our main motivation in this work is to construct a physically consistent data-driven SGS model that can be trained in a data-sparse regime and is also stable in the *a posteriori* simulation. We assess the performance of our *a posteriori* simulation using several statistical metrics and compare it with the statistics from filtered DNS solution. The turbulent kinetic energy at time t_k is computed as follows

$$TKE(t_k) = \lambda(u_f^2(t_k) + v_f^2(t_k)), \quad (28)$$

where u_f and v_f are the fluctuating components of velocity given by

$$u_f = \bar{u} - \lambda(\bar{u}), \quad (29)$$

$$v_f = \bar{v} - \lambda(\bar{v}), \quad (30)$$

where $\lambda(a)$ represents the spatial average of the field variable a . The velocity \bar{u} , and \bar{v} are computed by spectral differentiation of streamfunction as shown below

$$\bar{u} = \frac{\partial \bar{\psi}}{\partial y}; \quad \bar{v} = -\frac{\partial \bar{\psi}}{\partial x}. \quad (31)$$

The vorticity variance at each time step is computed as

$$\sigma^2 = \lambda((\bar{\omega} - \lambda(\bar{\omega}))^2). \quad (32)$$

We compare the kinetic-energy spectra and the vorticity structure function at intermediate time $t = 2.0$ and at final time $t = 4.0$ with the k^{-3} scaling which is observed in two-dimensional turbulence based on the classical KBL theory. The vorticity structure function is calculated using the formula given by [102] for two-dimensional turbulence as follows

$$S_\omega = \langle |\bar{\omega}(\mathbf{x} + \mathbf{r}) - \bar{\omega}(\mathbf{x})|^2 \rangle, \quad (33)$$

where $\langle \rangle$ indicates ensemble averaging, \mathbf{x} is the position on the grid, and \mathbf{r} is certain distance from this location. The PDF of the vorticity increment is utilized to assess the capability of SGS models in predicting the coherent vortices in the flow. The vorticity increments at different separations \mathbf{r} is defined as

$$\delta\omega(\mathbf{r}) = \omega(\mathbf{x} + \mathbf{r}) - \omega(\mathbf{r}). \quad (34)$$

We reiterate here that neural network-based SGS models are trained using the data for Reynolds number $Re = 16000$ and a single initial condition. Once the models are trained, the LES coupled with SGS models is performed for Reynolds number up to $Re = 128000$ and for five different initial conditions. Fig. 5 shows the evolution of turbulent kinetic energy and vorticity variance for LES runs with five different initial conditions and for several Reynolds numbers. For all the LES runs, we initialize the vorticity field at $t = 0.5$ after the initial spin-up period using the filtered DNS solution. We can observe that the model \mathcal{M}_{CNN} is stable only for short time and quickly becomes unstable after $t \approx 2.0$ even for Reynolds number $Re = 16000$ which was included in the training. In contrast to \mathcal{M}_{CNN} , model $\mathcal{M}_{\text{FL-CNN}}$ is stable for all test cases conducted here without any post-processing of the predicted SGS source term. The ensemble averaging procedure in DSM leads to highly dissipative results and is noticeable in the overprediction of the energy decay rate. The results of the LES runs with $\mathcal{M}_{\text{FL-CNN}}$ have the best agreement with filtered DNS solution for both turbulent kinetic energy and the vorticity variance.

Fig. 6 displays the kinetic-energy spectra at intermediate time $t = 2.0$ and at final time $t = 4.0$ obtained from LES runs with five different initial conditions for multiple Reynolds

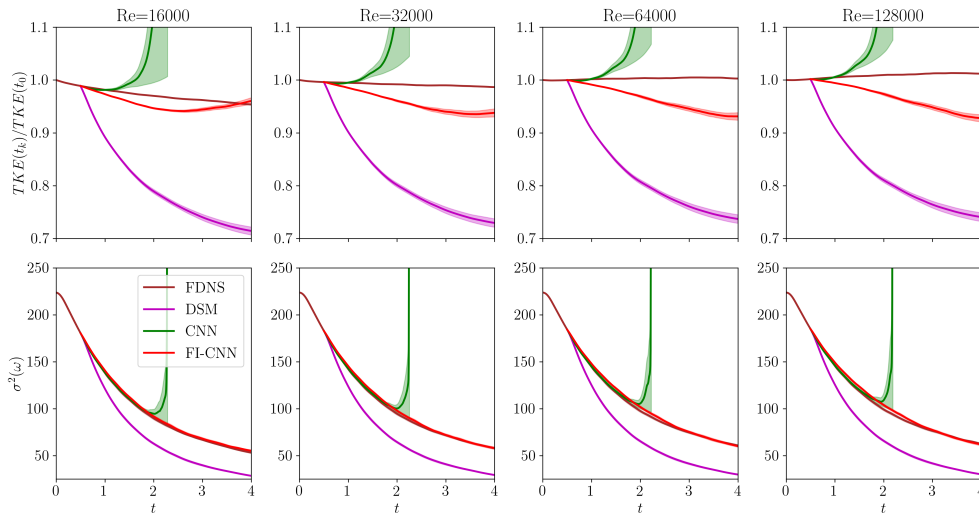


Figure 5: The time evolution of the turbulent kinetic energy $TKE(t_k)$ normalized by the initial turbulent kinetic energy $TKE(t_0)$ (top row) and vorticity variance (bottom row) for different Reynolds numbers at 256^2 grid resolution. The solid line shows the mean from LES runs for five different initial conditions and the shaded area corresponds to one standard deviation. The LES simulation starts at $t = 0.5$ after the initial spin-up time (i.e., once the turbulence has set in). The CNN and FI-CNN models are trained using the data generated from a single initial condition at Reynolds number $Re = 16000$.

number. Although the LES runs coupled with \mathcal{M}_{CNN} is stable at $t = 2.0$, the solution is unphysical as seen by the energy pile up near grid cutoff wavenumbers. This behavior is also illustrated in Fig. 7 through a large value of vorticity structure function at $t = 2.0$ across all Reynolds numbers. The LES runs with \mathcal{M}_{CNN} has diverged around $t \approx 2.5$ (see Fig. 5), and, therefore the kinetic-energy spectra and vorticity structure function are missing at $t = 4.0$ in Fig. 6 and Fig. 7, respectively. There is a very good agreement between the kinetic-energy spectra for LES runs with $\mathcal{M}_{\text{FL-CNN}}$ and filtered DNS solution, especially in the inertial subrange and k^{-3} theoretical scaling is captured accurately. From Fig. 7, we can see that the model $\mathcal{M}_{\text{FL-CNN}}$ is successful in producing the $r^{3/2}$ scaling [103] for the vorticity structure function at small scales and it gradually flattens near the large scales. The excessive dissipation of DSM is also illustrated in Fig. 6 and Fig. 7 via mismatch between kinetic-energy spectra and vorticity structure function between DSM and filtered DNS solution. The successful performance of LES runs with $\mathcal{M}_{\text{FL-CNN}}$ demonstrates that incorporating frame symmetries as hard constraints has been effective in stabilizing the coarse-grid simulation and in ensuring generalized learning across different initial conditions and Reynolds numbers.

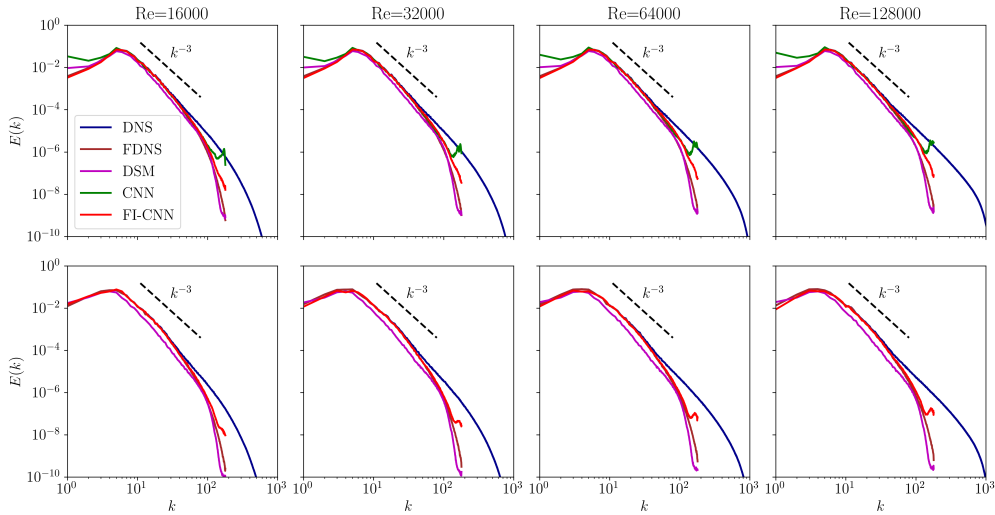


Figure 6: *A posteriori* kinetic-energy spectra for different Reynolds numbers at $t = 2.0$ (top row) and $t = 4.0$ (bottom row). These results are obtained from LES runs with five different initial conditions and only mean kinetic energy spectrum is shown. Note here that the CNN model has diverged and the kinetic-energy spectra for the CNN model is missing at the final time $t = 4.0$ (bottom row).

Figs. 8-11 provides the visualization of vorticity field and probability density function (PDF) of vorticity increments for Reynolds number $\text{Re} = 16000$ to $\text{Re} = 128000$ computed using the filtered DNS solution, LES with DSM model, and LES with $\mathcal{M}_{\text{FL-CNN}}$ at final time $t = 4.0$. We remark here that these results correspond to only one initial condition

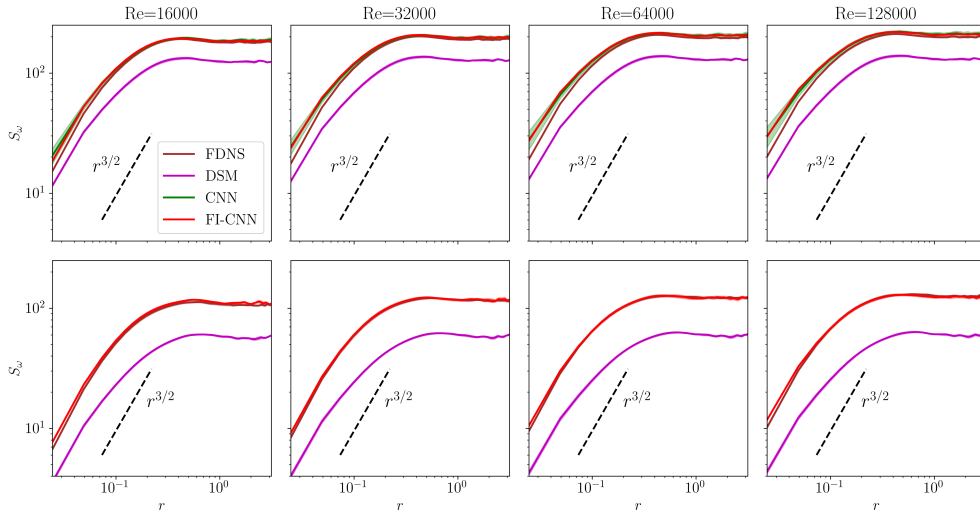


Figure 7: *A posteriori* second-order vorticity structure for different Reynolds numbers at $t = 2.0$ (top row) and $t = 4.0$ (bottom row). These results are obtained from LES runs with five different initial conditions and the solid line shows the mean vorticity structure and the shaded area corresponds to one standard deviation. Note here that the CNN model has diverged and the vorticity structure for the CNN model is not present at the final time $t = 4.0$ (bottom row).

that is different from the one used for training the neural network-based SGS models. Even though the LES with DSM model is successful in capturing large-scale structures in the flow, it fails to capture the small-scale structure due to excessive dissipation. The LES with $\mathcal{M}_{\text{FI-CNN}}$ is able to capture both large- and small-scale structures in the flow, and this can be ascertained to the stabilizing property of $\mathcal{M}_{\text{FI-CNN}}$ in the *a posteriori* deployment without any post-processing of the predicted SGS source term. Qualitatively, the vorticity field obtained from LES with model $\mathcal{M}_{\text{FI-CNN}}$ is very similar to the filtered DNS solution. The similarity in the shape of the PDF of vorticity increments as shown in Figs. 8-11 suggests the scale-invariant statistics of turbulence at all Reynolds numbers investigated in this study. The shape of the PDF of vorticity increments predicted by the LES with $\mathcal{M}_{\text{FI-CNN}}$ matches with the shape of the filtered DNS solution, and the heavy exponential tails in the PDF are related to the presence of coherent vortices in the flow. These heavy exponential tails are missing in the PDF of the solution obtained from LES with DSM, and it follows the Gaussian distribution.

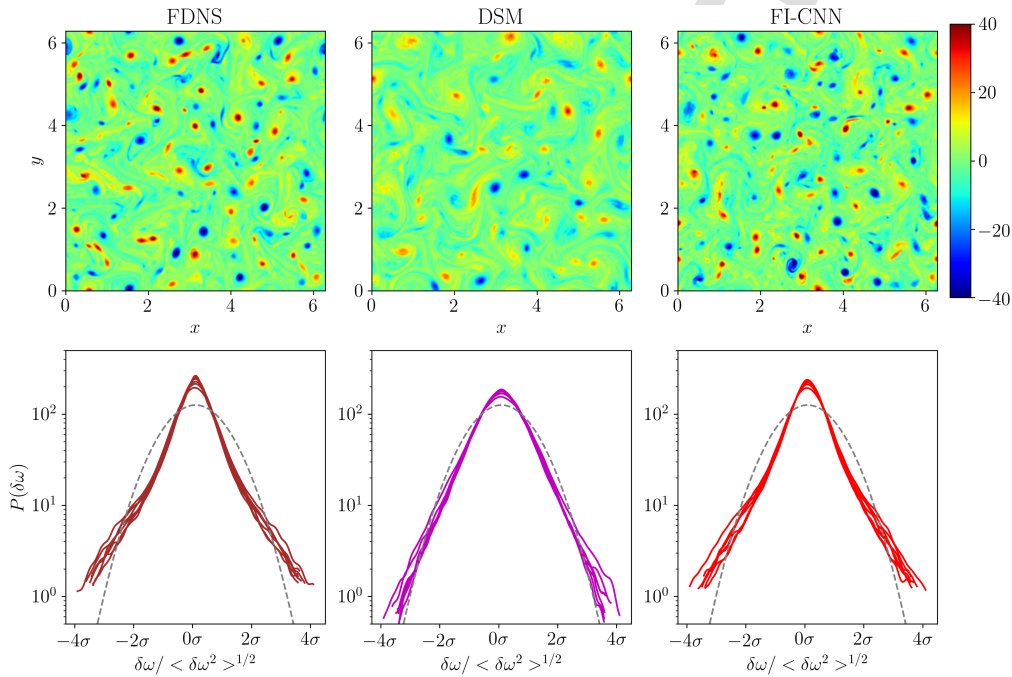


Figure 8: Snapshots of the vorticity distribution for $\text{Re} = 16000$ taken at final time $t = 4.0$ (top row) from different models and compared qualitatively against the FDNS solution. The bottom row displays the probability density function $P(\delta\omega)$ of the vorticity increments $\delta\omega$ for separations $r = 2\pi/256, 2\pi/128, 2\pi/64, 2\pi/32, 2\pi/16, 2\pi/8,$ and $2\pi/4$ computed from different models at $\text{Re} = 16000$. A Gaussian distribution is given in gray dashed line for comparison. We also note that the plain vanilla CNN becomes numerically unstable and unbounded before $t = 4.0$.

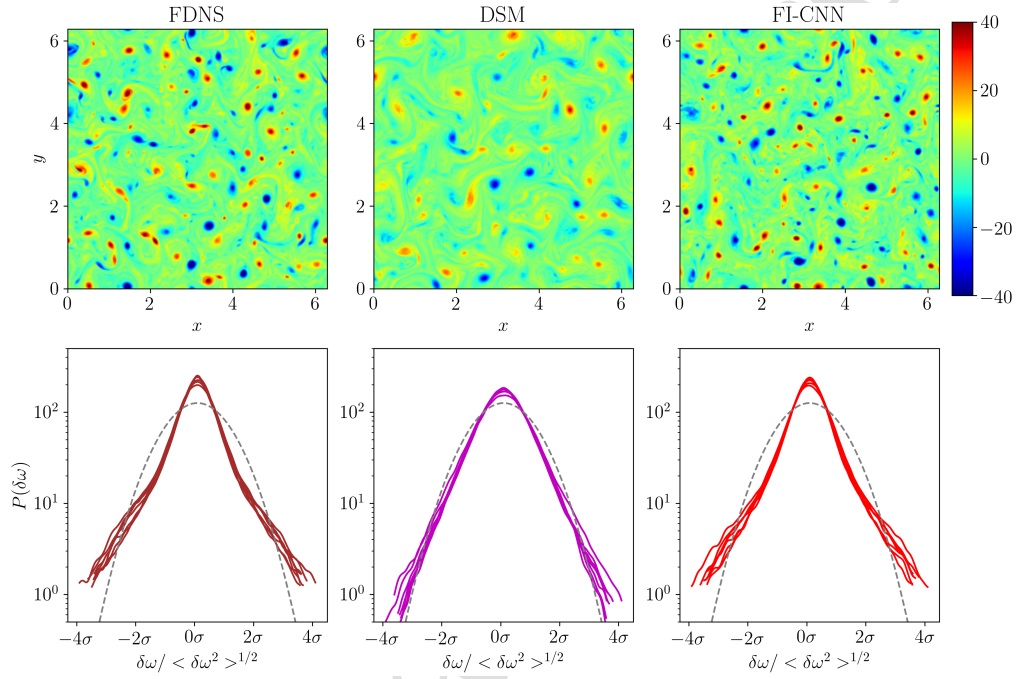


Figure 9: Snapshots of the vorticity distribution for $Re = 32000$ taken at final time $t = 4.0$ (top row) from different models and compared qualitatively against the FDNS solution. The bottom row displays the probability density function $P(\delta\omega)$ of the vorticity increments $\delta\omega$ for separations $r = 2\pi/256, 2\pi/128, 2\pi/64, 2\pi/32, 2\pi/16, 2\pi/8,$ and $2\pi/4$ computed from different models at $Re = 32000$. A Gaussian distribution is given in gray dashed line for comparison. We also note that the plain vanilla CNN becomes numerically unstable and unbounded before $t = 4.0$.

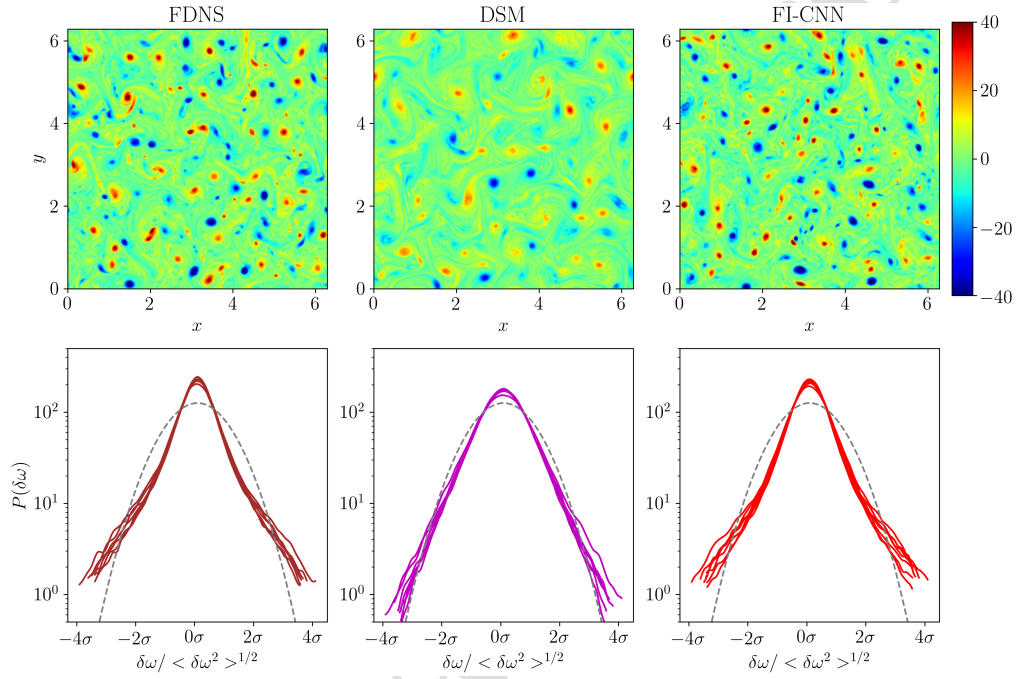


Figure 10: Snapshots of the vorticity distribution for $Re = 64000$ taken at final time $t = 4.0$ (top row) from different models and compared qualitatively against the FDNS solution. The bottom row displays the probability density function $P(\delta\omega)$ of the vorticity increments $\delta\omega$ for separations $r = 2\pi/256, 2\pi/128, 2\pi/64, 2\pi/32, 2\pi/16, 2\pi/8, \text{ and } 2\pi/4$ computed from different models at $Re = 64000$. A Gaussian distribution is given in gray dashed line for comparison. We also note that the plain vanilla CNN becomes numerically unstable and unbounded before $t = 4.0$.

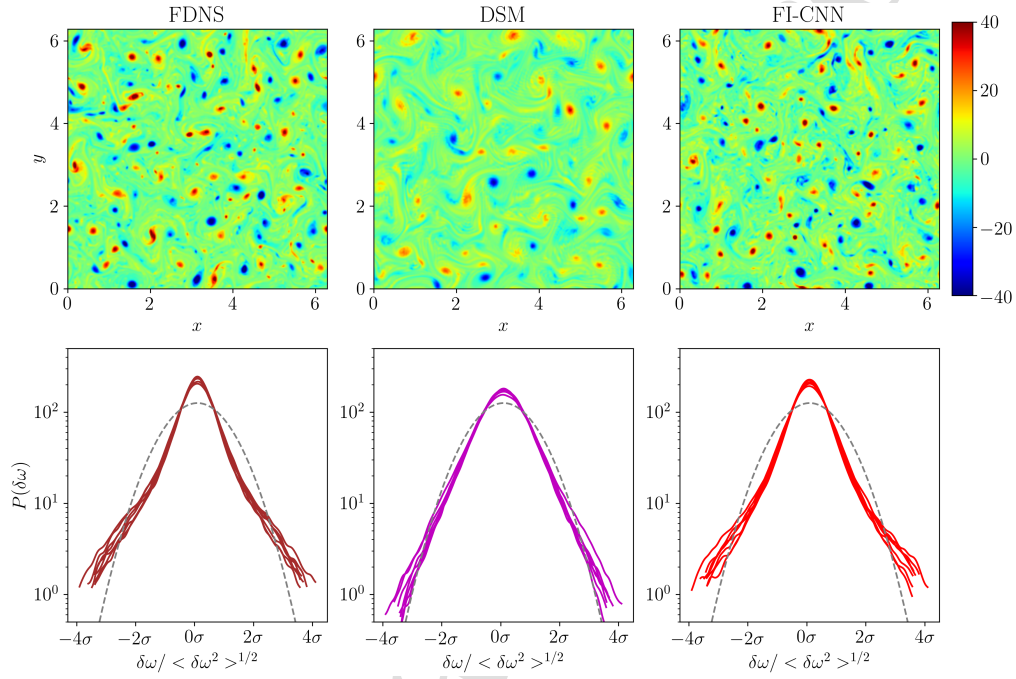


Figure 11: Snapshots of the vorticity distribution for $Re = 128000$ taken at final time $t = 4.0$ (top row) from different models and compared qualitatively against the FDNS solution. The bottom row displays the probability density function $P(\delta\omega)$ of the vorticity increments $\delta\omega$ for separations $r = 2\pi/256, 2\pi/128, 2\pi/64, 2\pi/32, 2\pi/16, 2\pi/8,$ and $2\pi/4$ computed from different models at $Re = 128000$. A Gaussian distribution is given in gray dashed line for comparison. We also note that the plain vanilla CNN becomes numerically unstable and unbounded before $t = 4.0$.

Next, we examine the robustness of the neural network-based SGS model by training an ensemble of neural networks using randomization-based approaches where different random initialization of weights are utilized for generating ensembles. Specifically, we train five neural networks for both models \mathcal{M}_{CNN} and $\mathcal{M}_{\text{FI-CNN}}$ using the same dataset as discussed in Section 4. This method is also applied to quantify the model-form uncertainty in deep learning [104]. Fig. 12 shows the time evolution of turbulent kinetic energy and vorticity variance at different Reynolds numbers. The time evolution of the TKE in Fig. 12 implies that the weights of the neural networks for $\mathcal{M}_{\text{FI-CNN}}$ are learned in such a way that the final models are overall dissipative in nature (as indicated by the solid line for mean from different LES runs). The vorticity variance predicted by the model $\mathcal{M}_{\text{FI-CNN}}$ is more accurate compared to DSM and is very close to the filtered DNS solution.

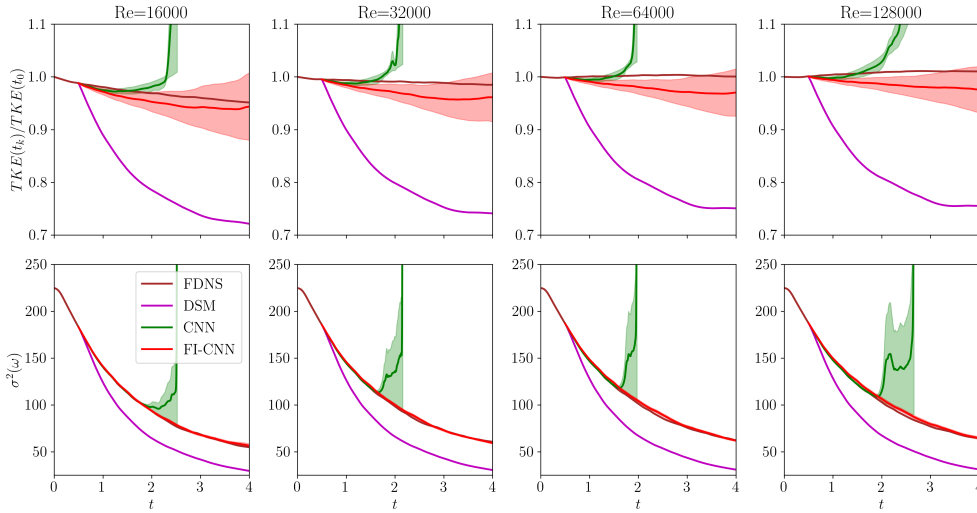


Figure 12: The time evolution of the turbulent kinetic energy $TKE(t_k)$ normalized by the initial turbulent kinetic energy $TKE(t_0)$ (top row) and vorticity variance (bottom row) for different Reynolds numbers at 256^2 grid resolution. The LES simulation starts at $t = 0.5$ after the initial spin-up time (i.e., once the turbulence has set in). For the CNN and FI-CNN models, an ensemble of neural networks is trained using the data generated from a single initial condition at Reynolds number $Re = 16000$ with different weights initialization. The solid line shows the mean from LES runs for a single initial condition with different trained networks and the shaded area corresponds to one standard deviation.

Fig. 13 depicts the kinetic-energy spectra at intermediate time $t = 2.0$ and at final time $t = 4.0$ obtained from LES runs for a single initial condition with different network-based SGS models for multiple Reynolds numbers. We observe the energy pile up near grid cutoff wavenumbers for the LES runs coupled with \mathcal{M}_{CNN} and this suggests that the solution is unphysical. This behavior is also demonstrated in Fig. 14 through a large value of vorticity structure function at $t = 2.0$ across all Reynolds numbers. The LES runs with \mathcal{M}_{CNN} has

diverged around $t \approx 2.5$ (as seen by large TKE in Fig. 12), and, therefore the kinetic-energy spectra and vorticity structure function are missing at $t = 4.0$ in Fig. 13 and Fig. 14, respectively. The kinetic-energy spectra for LES runs with $\mathcal{M}_{\text{FL-CNN}}$ is highly accurate and shows an excellent agreement with the filtered DNS solution, especially in the inertial subrange. The small uncertainty band also suggests that an ensemble of neural networks have produced very similar statistics for $\mathcal{M}_{\text{FL-CNN}}$. Fig. 14 shows that the model $\mathcal{M}_{\text{FL-CNN}}$ is successful in capturing the $r^{3/2}$ scaling for the vorticity structure function at small scales and flattening near large scales. With this numerical experiment, we can establish that the $\mathcal{M}_{\text{FL-CNN}}$ is robust, trustworthy, and stable in the LES, and it also ensures generalizable learning across different initial conditions and Reynolds numbers.

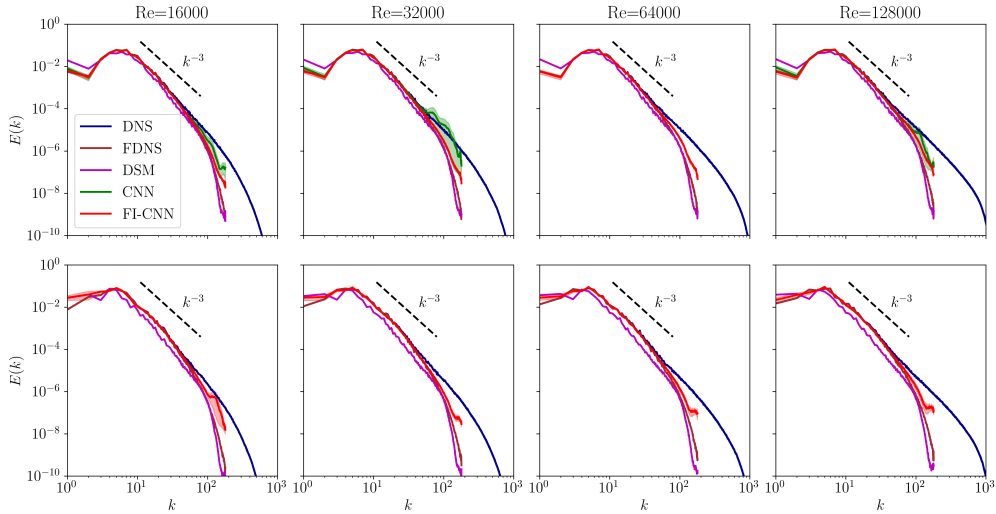


Figure 13: *A posteriori* kinetic-energy spectra for different Reynolds numbers at $t = 2.0$ (top row) and $t = 4.0$ (bottom row). The solid line shows the mean from LES runs for a single initial condition with different trained networks and the shaded area corresponds to one standard deviation. Note here that the CNN model has already diverged and the kinetic-energy spectra for the CNN model is missing at the final time $t = 4.0$ (bottom row).

6. Concluding remarks

Closure modeling in fluid dynamics simulations refers to parameterizing the interactions between high-fidelity and coarse-fidelity descriptions. In this study, we explore data-driven closure modeling strategies to improve both the accuracy and generalizability of such residual models. The motivation behind data-driven closure modeling stems from the fact that most of the existing SGS models are derived based on physical and mathematical considerations, and might not account for the important transfer of kinetic energy from small

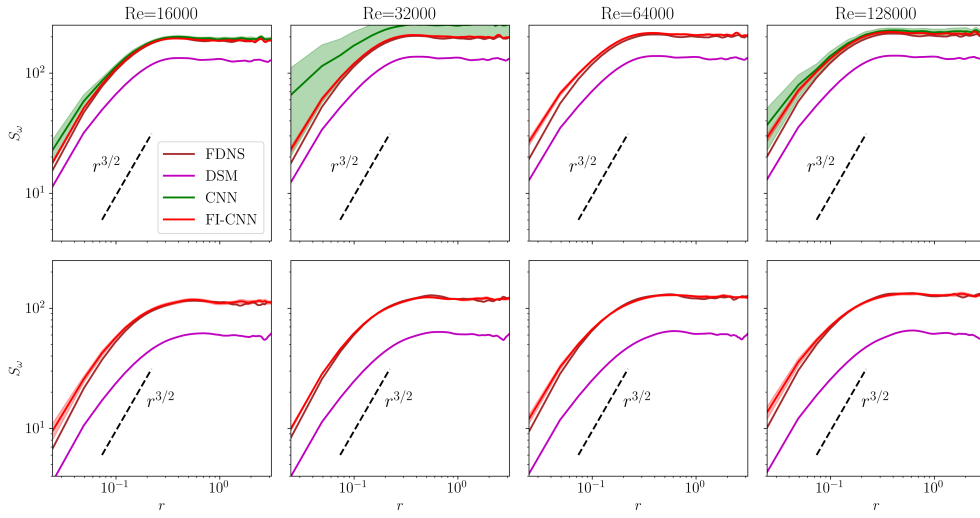


Figure 14: *A posteriori* second-order vorticity structure for different Reynolds numbers at $t = 2.0$ (top row) and $t = 4.0$ (bottom row). The solid line shows the mean from LES runs for a single initial condition with different trained networks and the shaded area corresponds to one standard deviation. Note here that the CNN model has diverged and the vorticity structure for the CNN model is not present at the final time $t = 4.0$ (bottom row).

360 scales to large scales (i.e., back-scatter) [105]. However, pure data-driven models can lead to physically inconsistent results [47] and might violate the known physical constraints. To address this limitation of pure data-driven models, we apply a frame invariant neural network architecture aiming at embedding physical symmetries directly into the structure of the convolutional neural networks. Thus, our model theoretically guarantees the frame symmetries, including translation, Galilean, and rotation invariance both during training and 365 inference. The embedding of physical symmetries as hard constraints not only improves the accuracy of the model but notably improves the generalization of the model, and eventually makes the model stable in their *a posteriori* deployment without any clipping.

We test the proposed framework for subgrid-scale modeling of Kraichnan turbulence in 370 *a priori* and *a posteriori* settings. The performance of the proposed framework is evaluated using several metrics like kinetic energy spectra, vorticity structure, and vorticity increments. Based on our analysis, we concluded that symmetry preservation has the potential to improve the accuracy, generalizability, and stability of the SGS model, besides embedding important geometric properties of the underlying PDEs into deep learning models. This work also 375 illustrates a broader lesson on how to combine machine learning with physics for scientific computing. It may be argued that two-dimensional turbulence is far from reality. However, it is generally considered as a canonical testbed for geophysical turbulence in the atmosphere and oceans. Our future development will be focused on scaling up the proposed frame invariant closure modeling framework to solve more realistic three-dimensional turbulent 380 flows. Another interesting avenue is to apply this framework for learning parameterization models for realistic geophysical flows with stratification, the Earth rotation, and Beta plane effects, paving the way for improved weather and climate prediction.

Data availability

The data that supports the findings of this study are available within the article. Implementation details and Python scripts can be accessed from the Github repository [106]. 385

Acknowledgements

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Award Number DE-SC0019290. O.S. gratefully acknowledges their support. Direct numerical simulations and 390 neural network training for this project was carried out using resources of the Oklahoma State University High-Performance Computing Center.

Disclaimer: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any 395 legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute

or imply its endorsement, recommendation, or favoring by the United States Government or
 400 any agency thereof. The views and opinions of authors expressed herein do not necessarily
 state or reflect those of the United States Government or any agency thereof.

References

- [1] S. B. Pope, S. B. Pope, Turbulent flows, Cambridge university press, 2000.
- 405 [2] P. Sagaut, Large eddy simulation for incompressible flows: an introduction, Springer Science & Business Media, 2006.
- [3] L. C. Berselli, T. Iliescu, W. J. Layton, Mathematics of Large Eddy Simulation of Turbulent Flows, Scientific Computation, Springer-Verlag, Berlin, 2006.
- [4] C. Meneveau, J. Katz, Scale-invariance and turbulence models for large-eddy simulation, Annual Review of Fluid Mechanics 32 (1) (2000) 1–32.
- 410 [5] K. Duraisamy, G. Iaccarino, H. Xiao, Turbulence modeling in the age of data, Annual Review of Fluid Mechanics 51 (2019) 357–377.
- [6] A. Beck, M. Kurz, A perspective on machine learning methods in turbulence modeling, GAMM-Mitteilungen 44 (1) (2021) e202100002.
- [7] S. L. Brunton, B. R. Noack, P. Koumoutsakos, Machine learning for fluid mechanics, Annual Review of Fluid Mechanics 52.
- 415 [8] M. Brenner, J. Eldredge, J. Freund, Perspective on machine learning for advancing fluid mechanics, Physical Review Fluids 4 (10) (2019) 100501.
- [9] F. Sarghini, G. De Felice, S. Santini, Neural networks based subgrid scale modeling in large eddy simulations, Computers & Fluids 32 (1) (2003) 97–108.
- 420 [10] A. Pal, Deep learning parameterization of subgrid scales in wall-bounded turbulent flows, arXiv preprint arXiv:1905.12765.
- [11] S. Pawar, O. San, A. Rasheed, P. Vedula, A priori analysis on deep learning of subgrid-scale parameterizations for Kraichnan turbulence, Theoretical and Computational Fluid Dynamics (2020) 387–401.
- [12] R. Maulik, O. San, J. D. Jacob, C. Crick, Sub-grid scale model classification and blending through deep learning, Journal of Fluid Mechanics 870 (2019) 784–812.
- 425 [13] G. Novati, H. L. de Laroussilhe, P. Koumoutsakos, Automating turbulence modelling by multi-agent reinforcement learning, Nature Machine Intelligence 3 (1) (2021) 87–96.
- [14] M. Gamahara, Y. Hattori, Searching for turbulence models by artificial neural network, Physical Review Fluids 2 (5) (2017) 054604.
- 430 [15] R. Maulik, O. San, A. Rasheed, P. Vedula, Subgrid modelling for two-dimensional turbulence using neural networks, Journal of Fluid Mechanics 858 (2019) 122–144.
- [16] C. Xie, J. Wang, K. Li, C. Ma, Artificial neural network approach to large-eddy simulation of compressible isotropic turbulence, Physical Review E 99 (5) (2019) 053113.
- [17] Y. Wang, Z. Yuan, C. Xie, J. Wang, Artificial neural network-based spatial gradient models for large-eddy simulation of turbulence, AIP Advances 11 (5) (2021) 055216.
- 435 [18] S. Rasp, M. S. Pritchard, P. Gentine, Deep learning to represent subgrid processes in climate models, Proceedings of the National Academy of Sciences 115 (39) (2018) 9684–9689.
- [19] D. J. Gagne, H. M. Christensen, A. C. Subramanian, A. H. Monahan, Machine learning for stochastic parameterization: Generative adversarial networks in the lorenz’96 model, Journal of Advances in Modeling Earth Systems 12 (3) (2020) e2019MS001896.
- 440 [20] S. Pawar, O. San, A. Rasheed, I. M. Navon, A nonintrusive hybrid neural-physics modeling of incomplete dynamical systems: Lorenz equations, GEM - International Journal on Geomathematics.
- [21] S. Pawar, O. San, Data assimilation empowered neural network parametrizations for subgrid processes in geophysical flows, Physical Review Fluids 6 (5) (2021) 050501.
- 445 [22] Y. Guan, A. Chattopadhyay, A. Subel, P. Hassanzadeh, Stable a posteriori les of 2d turbulence using convolutional neural networks: Backscattering analysis and generalization to higher re via transfer learning, arXiv preprint arXiv:2102.11400.

- [23] A. Beck, D. Flad, C.-D. Munz, Deep neural networks for data-driven les closure models, *Journal of Computational Physics* 398 (2019) 108910.
- 450 [24] T. Bolton, L. Zanna, Applications of deep learning to ocean data inference and subgrid parameterization, *Journal of Advances in Modeling Earth Systems* 11 (1) (2019) 376–399.
- [25] H. Frezat, G. Balarac, J. Le Sommer, R. Fablet, R. Lguensat, Physical invariance in neural networks for subgrid-scale scalar flux modeling, *Physical Review Fluids* 6 (2) (2021) 024607.
- [26] R. Maulik, O. San, A. Rasheed, P. Vedula, Data-driven deconvolution for large eddy simulations of kraichnan turbulence, *Physics of Fluids* 30 (12) (2018) 125109.
- 455 [27] Z. Yuan, C. Xie, J. Wang, Deconvolutional artificial neural network models for large eddy simulation of turbulence, *Physics of Fluids* 32 (11) (2020) 115106.
- [28] K. Fukami, K. Fukagata, K. Taira, Super-resolution reconstruction of turbulent flows with machine learning, *Journal of Fluid Mechanics* 870 (2019) 106–120.
- 460 [29] H. Kim, J. Kim, S. Won, C. Lee, Unsupervised deep learning for super-resolution reconstruction of turbulence, *Journal of Fluid Mechanics* 910.
- [30] J.-L. Wu, H. Xiao, E. Paterson, Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework, *Physical Review Fluids* 3 (7) (2018) 074602.
- [31] E. J. Parish, K. Duraisamy, A paradigm for data-driven predictive modeling using field inversion and machine learning, *Journal of Computational Physics* 305 (2016) 758–774.
- 465 [32] V. Srivastava, K. Duraisamy, Generalizable physics-constrained modeling using learning and inference assisted by feature space engineering, arXiv preprint arXiv:2103.16042.
- [33] P. Wu, X. Chang, W. Yuan, J. Sun, W. Zhang, R. Arcucci, Y. Guo, Fast data Assimilation (FDA): Data assimilation by machine learning for faster optimize model state, *Journal of Computational Science* 51 (2021) 101323.
- 470 [34] R. Arcucci, Effective data assimilation with machine learning, in: *Data Science and Big Data Analytics in Smart Environments*, CRC Press, 2021, pp. 27–46.
- [35] J. Mack, R. Arcucci, M. Molina-Solana, Y.-K. Guo, Attention-based convolutional autoencoders for 3d-variational data assimilation, *Computer Methods in Applied Mechanics and Engineering* 372 (2020) 113291.
- 475 [36] C. Buizza, C. Quilodrán Casas, P. Nadler, J. Mack, S. Marrone, Z. Titus, C. Le Cornec, E. Heylen, T. Dur, L. Baca Ruiz, C. Heaney, J. A. Díaz Lopez, K. S. Kumar, R. Arcucci, Data learning: Integrating data assimilation and machine learning, *Journal of Computational Science* 58 (2022) 101525. doi:<https://doi.org/10.1016/j.jocs.2021.101525>.
- 480 [37] M. Cheng, F. Fang, C. C. Pain, I. M. Navon, Data-driven modelling of nonlinear spatio-temporal fluid flows using a deep convolutional generative adversarial network, *Computer Methods in Applied Mechanics and Engineering* 365 (2020) 113000.
- [38] M. Cheng, F. Fang, C. Pain, I. M. Navon, An advanced hybrid deep adversarial autoencoder for parameterized nonlinear fluid flow modelling, *Computer Methods in Applied Mechanics and Engineering* 372 (2020) 113375.
- 485 [39] M. Cheng, F. Fang, I. M. Navon, C. Pain, A real-time flow forecasting with deep convolutional generative adversarial network: Application to flooding event in Denmark, *Physics of Fluids* 33 (5) (2021) 056602.
- [40] D. Xiao, F. Fang, C. Heaney, I. Navon, C. Pain, A domain decomposition method for the non-intrusive reduced order modelling of fluid flow, *Computer Methods in Applied Mechanics and Engineering* 354 (2019) 307–330.
- 490 [41] D. Xiao, C. Heaney, L. Mottet, F. Fang, W. Lin, I. Navon, Y. Guo, O. Matar, A. Robins, C. Pain, A reduced order model for turbulent flows in the urban environment using machine learning, *Building and Environment* 148 (2019) 323–337.
- [42] P. Wu, J. Sun, X. Chang, W. Zhang, R. Arcucci, Y. Guo, C. C. Pain, Data-driven reduced order model with temporal convolutional neural network, *Computer Methods in Applied Mechanics and Engineering* 360 (2020) 112766.
- 495 [43] R. Arcucci, C. Q. Casas, D. Xiao, L. Mottet, F. Fang, P. Wu, C. Pain, A Domain Decomposition

- Reduced Order Model with Data Assimilation (DD-RODA), IOS Press BV, 2020.
- 500 [44] S. E. Ahmed, S. Pawar, O. San, A. Rasheed, T. Iliescu, B. R. Noack, On closures for reduced order models—a spectrum of first-principle to machine-learned avenues, *Physics of Fluids* 33 (9) (2021) 091301.
- [45] J. Wu, H. Xiao, R. Sun, Q. Wang, Reynolds-averaged Navier–Stokes equations with explicit data-driven Reynolds stress closure can be ill-conditioned, *Journal of Fluid Mechanics* 869 (2019) 553–586.
- 505 [46] B. Nadiga, D. Livescu, Instability of the perfect subgrid model in implicit-filtering large eddy simulation of geostrophic turbulence, *Physical Review E* 75 (4) (2007) 046303.
- [47] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, *Nature Reviews Physics* 3 (6) (2021) 422–440.
- [48] K. Kashinath, M. Mustafa, A. Albert, J. Wu, C. Jiang, S. Esmailzadeh, K. Azizzadenesheli, R. Wang, A. Chattopadhyay, A. Singh, et al., Physics-informed machine learning: case studies for weather and climate modelling, *Philosophical Transactions of the Royal Society A* 379 (2194) (2021) 20200093.
- 510 [49] A. Prakash, K. E. Jansen, J. A. Evans, Invariant data-driven subgrid stress modeling in the strain-rate eigenframe for large eddy simulation, arXiv preprint arXiv:2106.13410.
- [50] J. Ling, A. Kurzwski, J. Templeton, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, *Journal of Fluid Mechanics* 807 (2016) 155–166.
- 515 [51] A. T. Mohan, N. Lubbers, D. Livescu, M. Chertkov, Embedding hard physical constraints in neural network coarse-graining of 3D turbulence, arXiv preprint arXiv:2002.00021.
- [52] A. Chattopadhyay, M. Mustafa, P. Hassanzadeh, E. Bach, K. Kashinath, Towards physically consistent data-driven weather forecasting: Integrating data assimilation with equivariance-preserving spatial transformers in a case study with era5, *Geoscientific Model Development Discussions* (2021) 1–23.
- 520 [53] S. Kneer, T. Sayadi, D. Sipp, P. Schmid, G. Rigas, Symmetry-aware autoencoders: s-pca and s-nlpca, arXiv preprint arXiv:2111.02893.
- [54] R. Wang, R. Walters, R. Yu, Incorporating symmetry into deep dynamics models for improved generalization, in: *International Conference on Learning Representations*, 2021.
- 525 [55] B. Siddani, S. Balachandar, R. Fang, Rotational and reflectional equivariant convolutional neural network for data-limited applications: Multiphase flow demonstration, *Physics of Fluids* 33 (10) (2021) 103323.
- [56] Y. Guan, A. Subel, A. Chattopadhyay, P. Hassanzadeh, Learning physics-constrained subgrid-scale closures in the small-data regime for stable and accurate LES, arXiv preprint arXiv:2201.07347.
- 530 [57] T. Cohen, M. Welling, Group equivariant convolutional networks, in: *International conference on machine learning*, PMLR, 2016, pp. 2990–2999.
- [58] M. Weiler, M. Geiger, M. Welling, W. Boomsma, T. Cohen, 3d steerable cnns: Learning rotationally equivariant features in volumetric data, arXiv preprint arXiv:1807.02547.
- [59] G. Boffetta, R. E. Ecke, Two-dimensional turbulence, *Annual Review of Fluid Mechanics* 44 (2012) 427–451.
- 535 [60] F. Bouchet, A. Venaille, Statistical mechanics of two-dimensional and geophysical flows, *Physics Reports* 515 (5) (2012) 227–295.
- [61] O. San, A. E. Staples, High-order methods for decaying two-dimensional homogeneous isotropic turbulence, *Computers & Fluids* 63 (2012) 105–127.
- 540 [62] O. San, A. E. Staples, Stationary two-dimensional turbulence statistics using a Markovian forcing scheme, *Computers & Fluids* 71 (2013) 1–18.
- [63] O. San, A dynamic eddy-viscosity closure model for large eddy simulations of two-dimensional decaying turbulence, *International Journal of Computational Fluid Dynamics* 28 (6-10) (2014) 363–382.
- [64] E. Ozbenli, P. Vedula, Construction of invariant compact finite-difference schemes, *Physical Review E* 101 (2) (2020) 023303.
- 545 [65] E. Ozbenli, P. Vedula, High order accurate finite difference schemes based on symmetry preservation, *Journal of Computational Physics* 349 (2017) 376–398.
- [66] E. Ozbenli, P. Vedula, Numerical solution of modified differential equations based on symmetry preservation, *Physical Review E* 96 (6) (2017) 063304.

- 550 [67] Y. Bar-Sinai, S. Hoyer, J. Hickey, M. P. Brenner, Learning data-driven discretizations for partial
differential equations, *Proceedings of the National Academy of Sciences* 116 (31) (2019) 15344–15349.
- [68] M. Oberlack, Invariant modeling in large-eddy simulation of turbulence, *Annual Research Briefs* (1997)
3–22.
- 555 [69] U. Frisch, A. N. Kolmogorov, *Turbulence: the legacy of AN Kolmogorov*, Cambridge university press,
1995.
- [70] C. G. Speziale, Galilean invariance of subgrid-scale stress models in the large-eddy simulation of
turbulence, *Journal of Fluid Mechanics* 156 (1985) 55–62.
- [71] J. Smagorinsky, General circulation experiments with the primitive equations: I. the basic experiment,
Monthly Weather Review 91 (3) (1963) 99–164.
- 560 [72] C. Leith, Atmospheric predictability and two-dimensional turbulence, *Journal of the Atmospheric
Sciences* 28 (2) (1971) 145–161.
- [73] M. Germano, U. Piomelli, P. Moin, W. H. Cabot, A dynamic subgrid-scale eddy viscosity model,
Physics of Fluids A: Fluid Dynamics 3 (7) (1991) 1760–1765.
- [74] D. K. Lilly, A proposed modification of the Germano subgrid-scale closure method, *Physics of Fluids
A: Fluid Dynamics* 4 (3) (1992) 633–635.
- 565 [75] J. S. Frederiksen, S. M. Kepert, Dynamical subgrid-scale parameterizations from direct numerical
simulations, *Journal of the Atmospheric Sciences* 63 (11) (2006) 3006–3019.
- [76] G. Kirkil, J. Mirocha, E. Bou-Zeid, F. K. Chow, B. Kosović, Implementation and evaluation of dynamic
subfilter-scale stress models for large-eddy simulation using wrf, *Monthly Weather Review* 140 (1)
570 (2012) 266–284.
- [77] T. Iliescu, P. Fischer, Backscatter in the rational LES model, *Computers & Fluids* 33 (5-6) (2004)
783–790.
- [78] S. Stolz, N. A. Adams, L. Kleiser, An approximate deconvolution model for large-eddy simulation
with application to incompressible wall-bounded flows, *Physics of fluids* 13 (4) (2001) 997–1015.
- 575 [79] O. San, A. E. Staples, T. Iliescu, Approximate deconvolution large eddy simulation of a stratified
two-layer quasigeostrophic ocean model, *Ocean Modelling* 63 (2013) 1–20.
- [80] J. Bardina, J. Ferziger, W. Reynolds, Improved subgrid-scale models for large-eddy simulation, in:
13th fluid and plasmadynamics conference, 1980, p. 1357.
- [81] F. Sarghini, U. Piomelli, E. Balaras, Scale-similar models for large-eddy simulations, *Physics of Fluids
11 (6) (1999) 1596–1607.*
- 580 [82] W. Layton, R. Lewandowski, A simple and stable scale-similarity model for large eddy simulation:
energy balance and existence of weak solutions, *Applied Mathematics Letters* 16 (8) (2003) 1205–1209.
- [83] S. Liu, C. Meneveau, J. Katz, On the properties of similarity subgrid-scale models as deduced from
measurements in a turbulent jet, *Journal of Fluid Mechanics* 275 (1994) 83–119.
- 585 [84] R. Maulik, O. San, A stable and scale-aware dynamic modeling framework for subgrid-scale parame-
terizations of two-dimensional turbulence, *Computers & Fluids* 158 (2017) 11–38.
- [85] K. Duraisamy, Perspectives on machine learning-augmented reynolds-averaged and large eddy simu-
lation models of turbulence, *Physical Review Fluids* 6 (5) (2021) 050504.
- [86] J. Kim, W. Jung, H. Kim, J. Lee, CyCNN: a rotation invariant CNN using polar mapping and
590 cylindrical convolution layers, arXiv preprint arXiv:2007.10588.
- [87] M. Weiler, G. Cesa, General $e(2)$ -equivariant steerable cnns, arXiv preprint arXiv:1911.08251.
- [88] M. Weiler, F. A. Hamprecht, M. Storath, Learning steerable filters for rotation equivariant cnns, in:
Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 849–858.
- [89] T. S. Cohen, M. Welling, Steerable cnns, arXiv preprint arXiv:1612.08498.
- 595 [90] E. J. Bekkers, M. W. Lafarge, M. Veta, K. A. Eppenhof, J. P. Pluim, R. Duits, Roto-translation
covariant convolutional networks for medical image analysis, in: *International conference on medical
image computing and computer-assisted intervention*, Springer, 2018, pp. 440–448.
- [91] A. Arakawa, Computational design for long-term numerical integration of the equations of fluid motion:
Two-dimensional incompressible flow. Part I, *Journal of Computational Physics* 135 (2) (1997) 103–
600 114.

- [92] P. Orlandi, Fluid flow phenomena: a numerical toolkit, Vol. 55, Springer Science & Business Media, 2000.
- [93] R. H. Kraichnan, Inertial ranges in two-dimensional turbulence, *The Physics of Fluids* 10 (7) (1967) 1417–1423.
- 605 [94] G. K. Batchelor, Computation of the energy spectrum in homogeneous two-dimensional turbulence, *The Physics of Fluids* 12 (12) (1969) II-233.
- [95] L. Zanna, T. Bolton, Data-driven equation discovery of ocean mesoscale closures, *Geophysical Research Letters* 47 (17) (2020) e2020GL088376.
- [96] U. Piomelli, W. H. Cabot, P. Moin, S. Lee, Subgrid-scale backscatter in turbulent and transitional flows, *Physics of Fluids A: Fluid Dynamics* 3 (7) (1991) 1766–1771.
- 610 [97] R. S. Rogallo, P. Moin, Numerical simulation of turbulent flows, *Annual review of fluid mechanics* 16 (1) (1984) 99–137.
- [98] J. Von Hardenberg, J. McWilliams, A. Provenzale, A. Shchepetkin, J. Weiss, Vortex merging in quasi-geostrophic flows, *Journal of Fluid Mechanics* 412 (2000) 331–353.
- 615 [99] R. Maulik, O. San, J. D. Jacob, Spatiotemporally dynamic implicit large eddy simulation using machine learning classifiers, *Physica D: Nonlinear Phenomena* 406 (2020) 132409.
- [100] R. Stoffer, C. M. van Leeuwen, D. Podareanu, V. Codreanu, M. A. Veerman, M. Janssens, O. K. Hartogensis, C. C. van Heerwaarden, Development of a large-eddy simulation subgrid model based on artificial neural networks: a case study of turbulent channel flow, *Geoscientific Model Development* 14 (6) (2021) 3769–3788. doi:10.5194/gmd-14-3769-2021.
- 620 [101] Z. Zhou, G. He, S. Wang, G. Jin, Subgrid-scale model for large-eddy simulation of isotropic turbulent flows using an artificial neural network, *Computers & Fluids* 195 (2019) 104319.
- [102] S. Grossmann, P. Mertens, Structure functions in two-dimensional turbulence, *Zeitschrift für Physik B Condensed Matter* 88 (1) (1992) 105–116.
- 625 [103] W. Kramer, G. Keetels, H. Clercx, G. van Heijst, Structure-function scaling of bounded two-dimensional turbulence, *Physical Review E* 84 (2) (2011) 026310.
- [104] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, in: *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, Curran Associates Inc., Red Hook, NY, USA, 2017, p. 6405–6416.
- 630 [105] H. T. Hewitt, M. Roberts, P. Mathiot, A. Biastoch, E. Blockley, E. P. Chassignet, B. Fox-Kemper, P. Hyder, D. P. Marshall, E. Popova, et al., Resolving and parameterising the ocean mesoscale in earth system models, *Current Climate Change Reports* 6 (2020) 137–152.
- [106] S. Pawar, FI-CNN, <https://github.com/surajp92/FI-CNN> (2022).

Highlights

- We introduce a novel data-driven closure approach.
- Our approach considers frame invariant properties in embedding physical symmetries.
- The proposed framework improves the generalizability of deep learning closure models.

Journal Pre-proof