Physical invariance in neural networks for subgrid-scale scalar flux modeling

Hugo Frezat *
University Grenoble Alpes, CNRS UMR LEGI, Grenoble, France

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

Julien Le Sommer
University Grenoble Alpes, CNRS UMR IGE, Grenoble, France

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

Redouane Lguensat
Laboratoire des Sciences du Climat et de l'Environnement (LSCE), IPSL/CEA, Gif Sur Yvette, France
and LOCEAN-IPSL, Sorbonne Université, Institut Pierre Simon Laplace, Paris, France

(Received 12 October 2020; accepted 1 February 2021; published 22 February 2021; corrected 18 March 2021)

In this paper we present a new strategy to model the subgrid-scale scalar flux in a three-dimensional turbulent incompressible flow using physics-informed neural networks (NNs). When trained from direct numerical simulation (DNS) data, state-of-the-art neural networks, such as convolutional neural networks, may not preserve well-known physical priors, which may in turn question their application to real case-studies. To address this issue, we investigate hard and soft constraints into the model based on classical transformation invariances and symmetries derived from physical laws. From simulation-based experiments, we show that the proposed transformation-invariant NN model outperforms both purely data-driven ones as well as parametric state-of-the-art subgrid-scale models. The considered invariances are regarded as regularizers on physical metrics during the *a priori* evaluation and constrain the distribution tails of the predicted subgrid-scale term to be closer to the DNS. They also increase the stability and performance of the model when used as a surrogate during a large-eddy simulation. Moreover, the transformation-invariant NN is shown to generalize to regimes that have not been seen during the training phase.

DOI: 10.1103/PhysRevFluids.6.024607

I. INTRODUCTION

The transport of a scalar quantity within a flow arises as a key issue in many applications, ranging from atmosphere and ocean physics to fluid dynamics for industrial purposes. While the scalar transport equations are known, solving this problem is still a difficult numerical challenge due to the large range of motion scales encountered in turbulent flows. As such, the direct numerical simulation (DNS) of realistic applications is not yet possible and reduced-order frameworks such as the large-eddy simulation (LES) have gained interest. The principle of the LES is to only compute the large

*hugo.frezat@univ-grenoble-alpes.fr

scales of the flow and take into account the interaction of the smallest scales on the large-scale dynamics through a subgrid-scale (SGS) model.

Following the recent advances in deep learning for turbulence modeling [1–4], neural networks (NNs) are expected to be good candidates for formulating subgrid closures and offer a promising alternative to algebraic models. Initial works in this direction were initiated by the modeling of the SGS momentum closure in incompressible [5,6], two-dimensional [7], or compressible [8] turbulence, among others. Due to their computational advantages, recent studies have investigated convolutional neural networks (CNNs) [9,10]. They have shown that CNNs outperform algebraic models in specific applications, e.g., in the estimation of the subgrid-scale reaction rates [11] or the subgrid parametrization of ocean dynamics [12]. However, these models do lack of embedded physical laws and are often thought as "black box" in which our comprehension is limited. This has two major implications for the model; first, the SGS model is able to predict only a narrow range of flows that have been used during training and suffer from rather limited generalization (or extrapolation) capabilities. Then, and most importantly, the SGS model will situationally predict unrealistic SGS terms that do not follow the expected behavior from well-known physical laws. Explicit embedding of known invariance has already shown great success [13,14] when applied to Galilean invariance in Reynolds Averaged Navier Stokes (RANS) simulation.

This paper addresses physical invariances in neural networks to improve the interpretability and generalization properties of SGS NN models. Whereas physical priors may be embedded implicitly through a custom training loss function [15], we rather explore how to explicitly embed expected physical invariances within a NN architecture. Through a case study of different regimes of scalar mixing in three-dimensional homogeneous isotropic turbulence, we show that the resulting NN framework significantly outperforms both state-of-the-art algebraic and NN models.

The paper is organized as follows: In Sec. II we introduce the SGS closure term that arises from the filtered transport equation and the modeling problem. Section III describes the numerical setup and baseline models used in our experiments along with different metrics for the evaluation. Demonstrating why and how physical invariances can be embedded into a NN is presented in Sec. IV. Finally, the evaluation *a priori* and *a posteriori* are presented for three different regimes in Sec. V.

II. SGS SCALAR MODELING

In this section we introduce the incompressible Navier-Stokes equations that govern the flow dynamics,

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{u} + F_{\mathbf{u}}, \quad \nabla\cdot\mathbf{u} = 0, \tag{1}$$

with velocity $\mathbf{u} = (u_x, u_y, u_z)$, density ρ, pressure p, kinematic viscosity ν, and $F_{\mathbf{u}}$ an external forcing. We define the advection diffusion of a scalar Φ by its Schmidt number, molecular diffusivity $\kappa = \nu/\mathrm{Sc}$, and source term F_Φ:

$$\frac{\partial \Phi}{\partial t} + (\mathbf{u}\cdot\nabla)\Phi = \nabla\cdot(\kappa\nabla\Phi) + F_\Phi. \tag{2}$$

The idea behind LES is to apply a scale separation that filters out the smallest scales of the scalar field Φ such that

$$\overline{\Phi(x)} = \int_V \Phi(x_f)\, G(x_f - x)\, dx_f, \tag{3}$$

where the kernel G is a spectral cutoff kernel defined in spectral space by the Heaviside step function H

$$\hat{G}(k) = H(\pi/\Delta - |k|), \tag{4}$$

where Δ is the filter width. Applying the filter (3) on (2) yields the filtered advection-diffusion equation

$$\frac{\partial \overline{\Phi}}{\partial t} + (\overline{\mathbf{u}} \cdot \nabla)\overline{\Phi} = \nabla \cdot (\kappa \nabla \overline{\Phi}) + \overline{F}_\Phi + \nabla \cdot (\overline{\mathbf{u}\,\Phi} - \overline{\mathbf{u}}\,\overline{\Phi}) \tag{5}$$

in which the residual flux is defined exactly as

$$\mathbf{s} \equiv \overline{\mathbf{u}\,\Phi} - \overline{\mathbf{u}}\,\overline{\Phi}. \tag{6}$$

Thus, the unknown SGS term that appears in (5) is the divergence of the residual flux, $\nabla \cdot \mathbf{s}$. Building a model that closes such equations is usually done following *functional* or *structural* approaches [16]. The functional strategy is based on physical considerations of the residual flux and not on the SGS term directly. Most functional models are based on the concept of eddy viscosity (or eddy diffusivity for a scalar) as in the Smagorinsky model [17] and have been widely used in practice. While being limited in the range of dynamics that can be reproduced, these models have the advantage to be stable in *a posteriori* evaluations, i.e., when studying their time evolution in a LES. The structural strategy on the other side aims at obtaining the best local approximation of the SGS term using the known small-scale structures of the flow. Structural models are built on a mathematical basis, often using a Taylor series expansion [18] or scale-similarity assumptions [19]. These models yield the best performance in *a priori* evaluations but are known to be unstable in long-term evolution due to their incorrect predictions of the subgrid local dissipation. Many improvements on the functional strategy have been explored; e.g., adding a dynamic coefficient [20,21] extends the capabilities of the model to a wider range of regimes, while regularization schemes for the structural strategy give rise to mixed models [22] using combinations of ideas from functional modeling.

With the emergence of deep learning [23], a renewed interest has been given to learning-based framework, especially NNs, as an alternative to algebraic schemes; see Refs. [24,25]. NN models result in parametric SGS models, whose parameters are trained to minimize some error over a training set. However, to our knowledge, as illustrated in Sec. IV, such NN models do not convey expected physical properties (e.g., invariance properties), which greatly impact their actual range of applicability.

III. NUMERICAL SETTING

NN SGS models require the generation of a training data set, usually coming from DNS, from which model parameters are learnt. This data set should contain a range of dynamics that are expected to be reproduced by the SGS model. When dealing with SGS scalar turbulence modeling, we follow the hypothesis given by Batchelor [26], which states that the small scales of a scalar quantity in a turbulent flow at high Reynolds number are statistically similar. Thus, the SGS data set shall comprise all the scales up to the limit between the inertial subrange and the energy containing range. To be precise, in the scalar-variance spectrum [27,28] given by

$$E_\Phi(k) = C_\Phi \chi \epsilon^{-1/3} k^{-5/3}, \tag{7}$$

where C_Φ is the Oboukou-Corrsin constant, and χ and ϵ are the scalar-variance and kinetic-energy dissipation rate, respectively, in spectral space at wave number k. We want to model the part of the spectrum that follows the $k^{-5/3}$ exponential form, i.e., in the inertial-convective subrange, which is known to be statistically universal within a considered turbulence regime.

A. Data generation

Data used in this work are generated from a DNS of a forced homogeneous isotropic turbulence. A classical pseudospectral code with second-order explicit Runge-Kutta time advancement is used to solve Eqs. (1) and (2) in a triply periodic domain $[0, 2\pi)^3$. The size of the computational domain

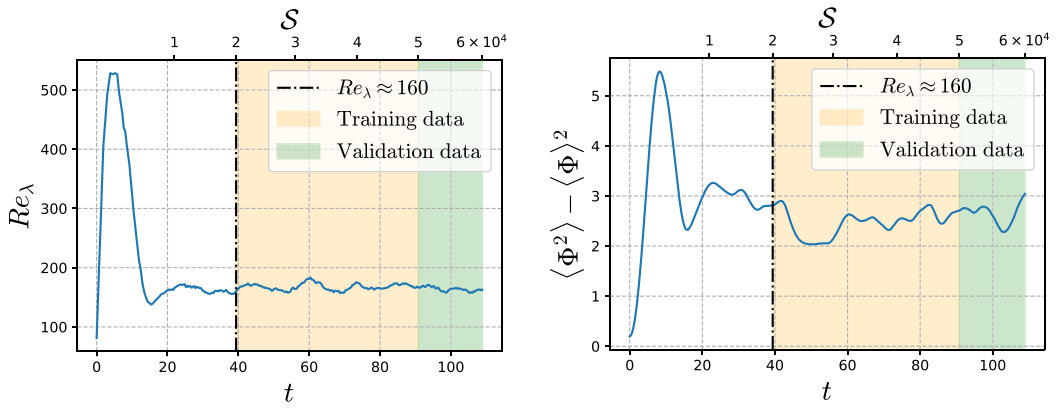

FIG. 1. Evolution of the Reynolds number on the Taylor microscale (left) and scalar variance (right). The flow is considered in an established turbulence regime at iteration $\mathcal{S} = 20\,000$. The first 50 samples are used for training, and the remaining 40 samples are used for validation.

is larger than four times the integral length scale to ensure that the largest flow structures are not affected, and the domain is discretized using 512^3 grid points. The simulation parameters are chosen such that $k_{\max}\eta > 1.5$ and $k_{\max}\eta_B > 1.5$, where $k_{\max}$ is the maximum wave number in the domain, and η and η_B are the Kolmogorov and Batchelor scales, respectively. The Reynolds number based on the Taylor microscale is around 160, and the molecular Schmidt number is set to 0.7. The initial random velocity field is generated in Fourier space as a solenoidal isotropic field with random phases and a prescribed energy spectrum as in Ref. [29], for example. The initial scalar field is a large-scale random field, generated according to the procedure proposed in Ref. [30]. Statistical stationarity is obtained using a random forcing located at low wave numbers, $k \in [2, 3]$, for both velocity [31] and scalar [32] fields. Applying the scalar spectral forcing only on the smallest wave numbers allows us to contain its effect to the resolved scales of the simulation. Note that the code has been intensively used for physical analysis of turbulent flows [33], numerical scheme development [34], and SGS model analysis [24,35].

Thus, we extract the filtered SGS term $\mathcal{M}_{\text{DNS}}$ and nonresidual input quantities that appears in (5),

$$\mathcal{M}_{\text{DNS}}(\mathbf{i}) = \boldsymbol{\nabla} \cdot \mathbf{s}, \quad \mathbf{i} = \{\overline{\mathbf{u}}, \overline{\Phi}\}. \tag{8}$$

To avoid large correlations in the data, we ensure that every sample is separated by almost one large eddy turnover time $t_L \sim L/U \sim (L^2/\epsilon)^{1/3}$, with L the integral scale, from previous and following ones and construct a data set from 80 equally spaced samples taken from 60 000 iterations. The learning data set is then split into two parts so that the 60 first samples are dedicated to training and the remaining 20 samples are used to validate the model, which is useful especially to monitor the behavior of the learning phase. For the test data set, we follow the same methodology and extract three different flow regimes used for the *a priori* evaluation; a developed turbulence statistically similar to the training data, where both velocity and scalar source terms are active, a forced transitional regime to turbulence with fields initialized at large scales, and a scalar decay driven by a forced turbulent flow. Reynolds number and scalar variance evolution as well as data separation from the data sets are shown in Figs. 1 and 2 for training, validation, and test, respectively. In Fig. 3 we show 2D slices at $z = \pi$ for different iterations $\mathcal{S}$ taken from the training data set.

B. Baseline models

For benchmarking purposes, we consider two algebraic closure models ($\mathbf{s}_{\text{DynSMAG}}$, $\mathbf{s}_{\text{DynRG}}$) and a recent model based on machine learning ($\mathbf{s}_{\text{MLP}}$). Both algebraic models are based on a dynamic

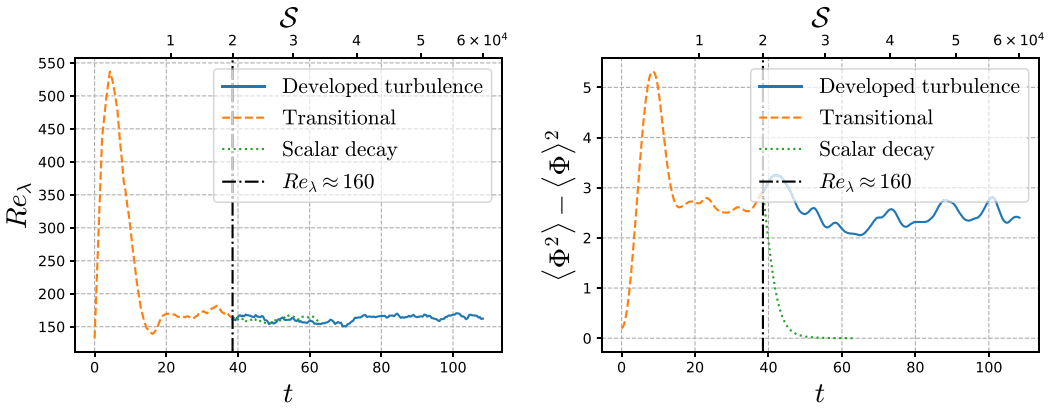

FIG. 2. Evolution of the Reynolds number on the Taylor microscale (left) and scalar variance (right) in the different regimes (developed turbulence, transitional and scalar decay) for the testing data.

procedure using a test filter noted $\hat{\cdot}$ such that its filter length scale $\hat{\Delta} = 2\bar{\Delta}$. The coefficient C can then be determined using some information on the resolved scales with Lilly's method [21].

The dynamic eddy diffusivity model [36] is an extension of the Smagorinsky model [20] for scalar transport. This is a functional model based on an eddy diffusivity leading to

$$\mathbf{s}_{\text{DynSMAG}} = C\bar{\Delta}^2||\bar{S}||\frac{\partial\overline{\Phi}}{\partial x_j}, \tag{9}$$

where $||\overline{S}||$ is the L_2 norm of the resolved strain rate tensor. This model is commonly used because it is stable when $C \in \mathbb{R}^*_+$, i.e., the SGS scalar dissipation rate, $\mathbf{s}_{\text{DynSMAG}} \cdot \nabla\overline{\Phi}$, is always positive, meaning that the scalar-variance transfers are always from large to small scales.

The gradient model on the other hand is a structural model based on a truncated series expansion using the filter scale [37]. This model is known to provide a good local approximation and correlation with the exact residual flux as Δ tends to zero [32]. However, the model is also producing back-scatter, i.e., transfers from modeled to resolved scales, which leads to instabilities due to the under-prediction of dissipation. To overcome this limitation, this model is often used in combination with an eddy diffusivity model [18]. Another recent improvement consists of a regularization of the model that eliminates back-scatter transfers [22]. The dynamic regularized gradient model then writes

$$\mathbf{s}_{\text{DynRG}} = C\bar{\Delta}^2\bar{S}^{\ominus}_{ij}\frac{\partial\overline{\Phi}}{\partial x_j}, \tag{10}$$

where $\bar{S}^{\ominus}_{ij}$ contains only the direct energy transfer from the strain rate tensor.

We also consider a recently published NN SGS model [25], which relies on a Multilayer Perceptron (MLP) to predict the residual flux. This model uses the gradients of grid quantities, $\overline{\mathbf{u}}$ and $\overline{\Phi}$ as inputs and optimizes L fully connected linear (or affine) layers with nonlinear ReLu activations $\mathcal{R}(x) = \max(0, x)$ such that

$$\mathbf{s}_{\text{MLP}} = \mathbf{W}^{(L)}\mathbf{h}^{(L-1)} + \mathbf{b}^{(L)}, \quad \mathbf{h}^{(i)} = \mathcal{R}(\mathbf{W}^{(i)}\mathbf{h}^{(i-1)} + \mathbf{b}^{(i)}), \quad \mathbf{h}^{(0)} = \{\overline{\mathbf{u}}, \overline{\Phi}\}, \tag{11}$$

where $\mathbf{b}$ is a bias vector and $\mathbf{W}$ a weight matrix. To our knowledge, this model is the closest to our work.

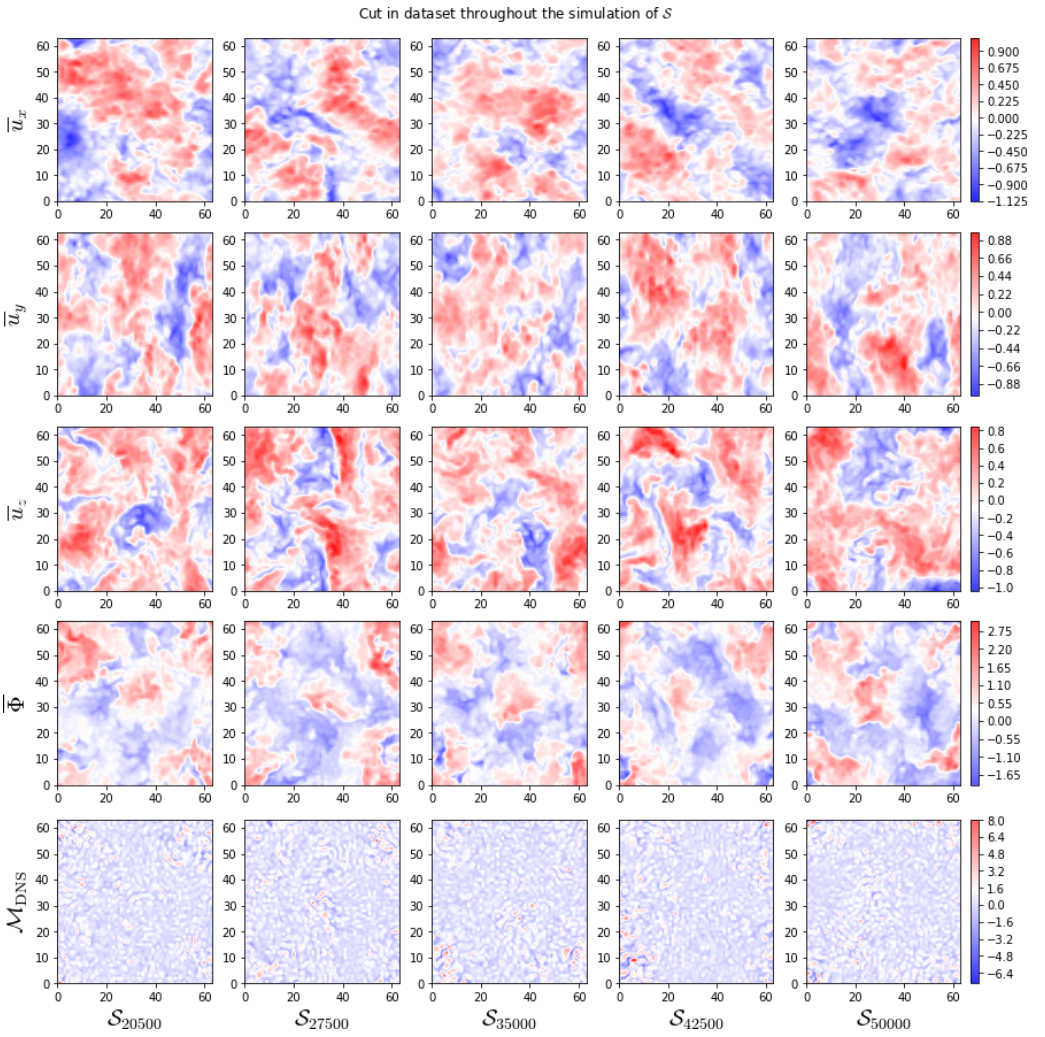

FIG. 3. Data slices extracted during the simulation. $\overline{\mathbf{u}}$ are the velocities in each directions, $\overline{\Phi}$ is the scalar quantity, and $\mathcal{M}_{\mathrm{DNS}}$ is the SGS term. The simulation was run with $N = 512$ grid points and filtered at $N = 64$. The flow reaches a statistically developed turbulence state approximately at iteration $\mathcal{S} = 20\,000$.

C. Metrics

Most *a priori* evaluations in the literature characterize the performance of a model based on a structural metric which is defined by the error on the SGS term and a functional metric that operates on the SGS residual flux. In particular, it is common to characterize the *a priori* performance of a given model by its ability to reproduce the distribution of SGS scalar dissipation, i.e., the transfer of energy between resolved and subgrid scales given by $\mathbf{s} \cdot \boldsymbol{\nabla}\overline{\Phi}$. In this paper, however, we focus on the modeling of the SGS term $\boldsymbol{\nabla} \cdot \mathbf{s}$, and thus our proposed model does not have access to the residual flux $\mathbf{s}$. Therefore, we perform an extensive study of the predicted SGS term (or structural performance) statistical properties. As discussed in Ref. [38], it is suggested that structural performance describes the short-term time evolution of a model and is of most importance to the understanding of local and instantaneous errors in a model. In this study, we evaluate three different types of metrics summarized in Table I based on structural, physical, and statistical considerations.

TABLE I. Equations for the structural, physical and statistical metrics used to evaluate the *a priori* performance of the different models.

Type	Metric	Equation
Structural metric	Normalized RMSE	$\mathcal{L}_{\text{rms}}(X,Y) = \sqrt{\frac{1}{N}\|\|X-Y\|\|_2}/\sqrt{\langle Y^2\rangle - \langle Y\rangle^2}$
	Pearson's coefficient	$\mathcal{P}(X,Y) = \text{cov}(X,Y)/\sigma_X\sigma_Y$
Physical metric	Integral dissipation error	$\mathcal{I}(X,Y) = -\int_V \overline{\Phi}\nabla\cdot\mathbf{s}_X\, dV + \int_V \overline{\Phi}\nabla\cdot\mathbf{s}_Y\, dV$
Statistical metric	Kullback-Leibler divergence	$\mathcal{D}(P_X\|\|P_Y) = \sum_i^N P_X(i)\ln[P_X(i)/P_Y(i)]$
	Jensen-Shannon distance	$\mathcal{J}(P_X\|\|P_Y) = \sqrt{\frac{1}{2}\mathcal{D}(P_X\|\|M) + \frac{1}{2}\mathcal{D}(P_Y\|\|M)}, M = \frac{1}{2}(P_X+P_Y)$
	Kolmogorov-Smirnov statistic	$\mathcal{K}(X,Y) = \sup_t \lvert F_X(t) - F_Y(t)\rvert, F_A(t) = \frac{1}{N}\sum_i^N \mathbf{1}_{A_i\leqslant t}$

The structural evaluation is given by the normalized root-mean-squared error (RMSE) $\mathcal{L}_{\text{rms}}$ and Pearson's cross-correlation coefficient $\mathcal{P}$ between the predicted SGS term and the filtered DNS. We also show the error on the integral of the SGS dissipation $\mathcal{I}$, which can be computed using the divergence of the SGS flux since

$$\int_V \mathbf{s}\cdot\nabla\overline{\Phi}\, dV = -\int_V \overline{\Phi}\nabla\cdot\mathbf{s}\, dV \tag{12}$$

if V is our periodic domain. Additionally, we consider statistical metrics to evaluate how consistent is the pdf of the predicted term w.r.t. the true one. The Jensen-Shannon distance $\mathcal{J}$ [39] is a metric that measures the similarity between two probability distributions P_X, P_Y defined on the same probability space $\mathcal{X}$ built on the Kullback-Leibler divergence $\mathcal{D}$ or relative entropy. Also, we compute a Kolmogorov-Smirnov statistic test $\mathcal{K}$ that hypothesizes that two samples are drawn from the same distributions if its value is small ($\leqslant 0.1$) as shown in Ref. [40].

IV. TRANSFORMATION-INVARIANT LEARNING FRAMEWORK

In this section we describe how we model the SGS term, $\nabla\cdot\mathbf{s}$, using a NN model, referred to as $\mathcal{M}_{\text{NN}}$, where the objective is to approximate

$$\mathcal{M}_{\text{NN}}(\overline{\mathbf{u}}, \overline{\Phi}) \approx \nabla\cdot\mathbf{s}. \tag{13}$$

We choose four nonexhaustive physical relationships that must be verified by any SGS scalar flux model and show how to include them within a NN as hard and soft constraints. The first three constraints are based on frame symmetry [41], which encompass Galilean, translation, and rotation invariance. The last constraint ensures the linearity of the transport equation, required in order to generalizes to different scalar quantities.

A. Translation invariance

Translation invariance states that the considered SGS NN model shall not be location-dependent. In other words, model $\mathcal{M}_{\text{NN}}$ shall satisfy the following constraint:

$$\forall\delta\in\mathbb{R},\quad \mathcal{M}_{\text{NN}}(T_\delta\overline{\mathbf{u}}, T_\delta\overline{\Phi}) = T_\delta\mathcal{M}_{\text{NN}}(\overline{\mathbf{u}}, \overline{\Phi}), \tag{14}$$

where T_δ is a translation operator for spatial displacement δ such that $T_\delta\mathbf{y}(\mathbf{x}) = \mathbf{y}(\mathbf{x}+\delta)$. In the deep learning literature, in contrast to fully connected architectures which refer to NN architectures based on dense or fully connected layers, e.g., the MLP used in Ref. [25], Convolutional Neural Networks (CNNs) are described by a combination of convolution and activation layers. Since the convolution commutes with translation, this type of architecture provides a built-in invariance to

TABLE II. Evaluation of the three additional constraints provided by $\mathcal{M}_{\text{SGTNN}}$; expectation and variance of the normalized root-mean-squared error on the SGS term predicted by NN models from many realizations on the testing data. Each row checks the transport linearity, Galilean, and rotation invariance, respectively. Note that λ and β are uniformly sampled values.

	$\mathcal{M}_{\text{CNN}}$		$\mathcal{M}_{\text{SGTNN}}$	
	$\text{E}[\mathcal{L}_{\text{rms}}]$	$\text{Var}[\mathcal{L}_{\text{rms}}]$	$\text{E}[\mathcal{L}_{\text{rms}}]$	$\text{Var}[\mathcal{L}_{\text{rms}}]$
$\mathcal{M}_{\text{NN}}(\overline{\mathbf{u}}, \lambda\overline{\Phi})$	1.0238	0.0945	0.8356	0.0
$\mathcal{M}_{\text{NN}}(\overline{\mathbf{u}} + \beta, \overline{\Phi})$	0.9330	0.0022	0.8356	0.0
$\mathcal{M}_{\text{NN}}(A_{ij}\overline{\mathbf{u}}, \overline{\Phi})$	0.8741	1.4540×10^{-7}	0.8355	2.044×10^{-8}

spatial displacements, which satisfy (14). We note that using a different set of inputs can already ensure some symmetries of the residual flux [42] but reduces the flexibility of the model.

Convolutional architectures also appear as a natural choice when dealing with structured tensors (i.e., time, space, space-time processes) as they relate to local filtering operations with respect to tensor dimensions. For instance, convolutional layers embed all discretized-filtering operations, such as first-order or higher-order derivatives of the considered variables. This may provide the basis for some physical interpretation of CNN architectures. Importantly, CNNs greatly outperform fully connected architectures and are currently the state-of-the-art schemes for a wide range of machine learning applications with one-dimensional [43], two-dimensional [44], or three-dimensional [45] states.

Let us call now $\mathcal{M}_{\text{CNN}}$ a model built from convolutions and nonlinear activation layers as presented above. This model is expected to perform well on the metric $\mathcal{L}$ for which it has been optimized but should not strictly or even numerically (as shown in Table II for some invariances except the translational one) verify well-established physical symmetries and properties. From now on, we will refer to the transformation-invariant NN model as $\mathcal{M}_{\text{SGTNN}}$ for "Subgrid Transport Neural Network."

B. Transport linearity

The linearity of the scalar transport equation is a fundamental property that shall be verified by the proposed SGS model. Formally, it comes to verify the following constraint:

$$\forall\lambda \in \mathbb{R}, \quad \mathcal{M}_{\text{NN}}(\overline{\mathbf{u}}, \lambda\overline{\Phi}) = \lambda\mathcal{M}_{\text{NN}}(\overline{\mathbf{u}}, \overline{\Phi}). \tag{15}$$

This relationship can not be verified within $\mathcal{M}_{\text{CNN}}$ since the model contains nonlinearities introduced by the $\mathcal{R}$ operator. We propose to split the model $\mathcal{M}_{\text{CNN}}$ into two submodels $\mathcal{V}$ and $\mathcal{T}$ that take as inputs the velocities and the transported scalar, respectively. $\mathcal{V}$ involves a classic CNN architecture. By contrast, to conform to the linearity property, model $\mathcal{T}$ applies a single convolution layer with no bias and nonlinear activation. The resulting SGS model is given by

$$\mathcal{M}_{\text{SGTNN}}(\overline{\mathbf{u}}, \overline{\Phi}) = [\mathcal{V}(\overline{\mathbf{u}}) \times \mathcal{T}(\overline{\Phi})] * \mathcal{C}^{+}, \tag{16}$$

where $*$ is the discrete convolution operator, $\times$ is a term-by-term matrix product, and $\mathcal{C}^{+}$ is the convolution kernel with unitary width $K = 1$. We sketch the resulting architecture in Fig. 4.

First, the periodic pad replicates boundaries according to the kernel width of convolutions kernels $\mathcal{C}$. The second stage applies in parallel operators $\mathcal{V}$ and $\mathcal{T}$ and computes the term-by-term product of their outputs, which are $128 \times 128 \times 128$ tensors. The last convolutional layer maps the resulting 128-dimensional representation to a scalar field using a kernel width of unitary size. We can now

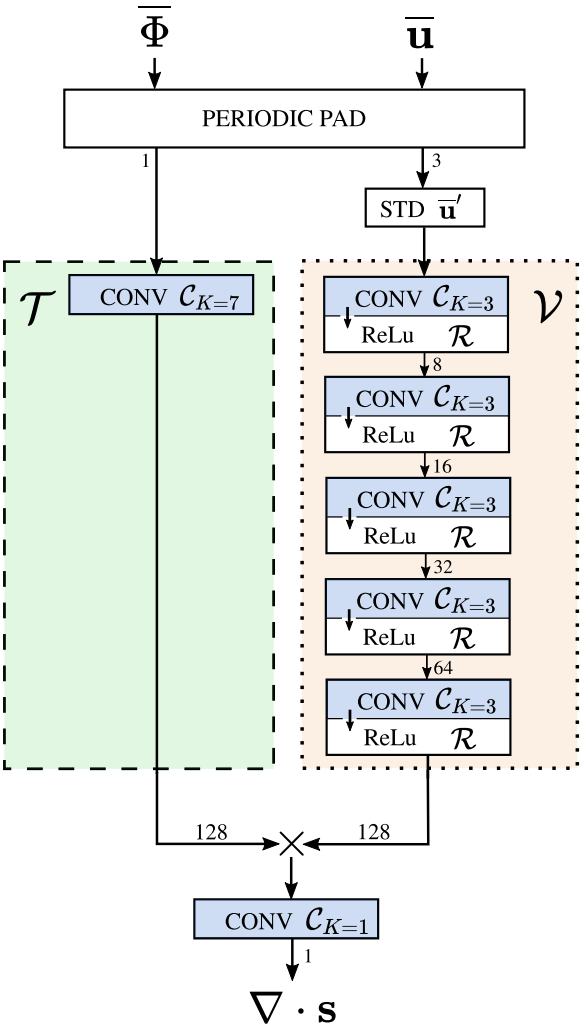

FIG. 4. Illustration of the subgrid transport neural network architecture (SGTNN). At the top, two non-learnable steps are applied; the periodic pad replicates boundaries according to the kernel width of convolution kernel and std transforms the velocities to ensures Galilean invariance. Then the two submodels $\mathcal{V}$ (right) for the velocity and $\mathcal{T}$ (left) for the scalar transport are represented in dotted and dashed lines, respectively. The convolution with kernel size $K = 1$ corresponding to $\mathcal{C}^+$ applied to the combined result of the latter is depicted at the bottom. The number of filters within the NN is shown along with the arrows.

show that the relationship (15) holds:

$$\mathcal{M}_{\text{SGTNN}}(\overline{\mathbf{u}}, \lambda\overline{\Phi}) = \sum_{i=0}^{d} \mathcal{V}(\overline{\mathbf{u}})_i \lambda \mathcal{T}(\overline{\Phi})_i C_i^+ = \lambda \sum_{i=0}^{d} \mathcal{V}(\overline{\mathbf{u}})_i \mathcal{T}(\overline{\Phi})_i C_i^+,$$

where d is the input dimension of $\mathcal{C}^+$ (here $d = 128$) and linearity of $\mathcal{T}$ is implied from its convolution. Finally,

$$\begin{aligned}\mathcal{M}_{\text{SGTNN}}(\overline{\mathbf{u}}, \lambda\overline{\Phi}) &= \lambda\{[\mathcal{V}(\overline{\mathbf{u}}) \times \mathcal{T}(\overline{\Phi})] * \mathcal{C}^+\} \\ &= \lambda \mathcal{M}_{\text{SGTNN}}(\overline{\mathbf{u}}, \overline{\Phi}),\end{aligned}$$

which validates that the architecture $\mathcal{M}_{\text{SGTNN}}$ has a linear path on $\overline{\Phi}$.

C. Galilean invariance

The next constraint is based on frame invariance. It is known that the Navier-Stokes and the transport equations as well as their filtered form are Galilean-invariant [42]. In order to fulfill this property on the modeled scales, we must ensure that our SGS model is also form-invariant under the Galilean group of transformations. In particular, we want to have a description of turbulence that is the same in all inertial frames of reference. This translates easily to our problem as

$$\forall \beta \in \mathbb{R}, \quad \mathcal{M}_{\mathrm{NN}}(\overline{\mathbf{u}} + \beta, \overline{\Phi}) = \mathcal{M}_{\mathrm{NN}}(\overline{\mathrm{u}}, \overline{\Phi}). \tag{17}$$

We propose to solve this problem by reducing the velocity to a standardized quantity, i.e, with zero mean $\langle(\overline{\mathbf{u}} + \beta)'\rangle = 0$ for each instantaneous sample. Let us denote the standardized operator $\cdot'$, and we have

$$\begin{aligned}(\overline{\mathbf{u}} + \beta)' &= \overline{\mathbf{u}} + \beta - \langle \overline{\mathbf{u}} + \beta \rangle \\ &= \overline{\mathbf{u}} - \langle \overline{\mathbf{u}} \rangle \\ &= (\overline{\mathbf{u}})'.\end{aligned}$$

Note that this assumption is true only if bias is removed from the operator $\mathcal{C}$. We include this standardization step after the periodic pad and apply it on the velocities prior to the submodel $\mathcal{V}$.

D. Rotation invariance

The last fundamental invariance required to fulfill frame symmetry is related to reflections and rotations [46]. The filtered equation (5) holds under the rotation of the coordinate system and velocity vector,

$$\mathcal{M}_{\mathrm{NN}}(A_{ij}\overline{\mathbf{u}}_j, \overline{\Phi}) = \mathcal{M}_{\mathrm{NN}}(\overline{\mathbf{u}}, \overline{\Phi}), \tag{18}$$

for any rotation matrix A with $A^T A = AA^T = I$ in a rotated frame $\mathbf{y}_i = A_{ij}\mathbf{y}_j$. Ensuring rotation invariance in a NN is a difficult problem since convolution kernels are not imposed but rather learnt by the optimization algorithm. Some progress has been made to exploit these symmetries [47,48] but are either limited to finite subsets of rotations or require major changes and priors in the architecture. A common and flexible way to approximate this invariance is to use data augmentation, i.e., extending the data set with new samples that exhibit the desired invariance. We used permutations of our initial data that verifies (18) with angles of 90°.

E. NN architecture and learning scheme

The optimization problem is formulated as the minimization of an energy term $\mathcal{L}$ also called a loss function, which in our case will be defined as a simple mean-squared error (L_2) on the SGS term prediction. Obtaining the gradients of $\mathcal{L}$ is done through automatic differentiation, which is available directly in most NN frameworks. Finally, the NN models optimize the parameter space w.r.t

$$\mathcal{L}[\mathcal{M}_{\mathrm{NN}}(\theta_k), \boldsymbol{\nabla} \cdot \mathbf{s}] = \|\mathcal{M}_{\mathrm{NN}}(\theta_k) - \boldsymbol{\nabla} \cdot \mathbf{s}\|_2, \tag{19}$$

where θ_k is the model input. Since our simulations are done in a periodic domain, we can replicate periodically our input at the boundaries without introducing any error. This allows us to remove any padding inside the NN.

After extensive experimentation, the best results in our scenarios were obtained using a velocity submodel $\mathcal{V}$ composed of six nonlinear convolution units, i.e., applying $\mathcal{R}$ after $\mathcal{C}$ with kernels of size $3 \times 3 \times 3$ and increasing the number of filters from 8 to 128. The transport submodel $\mathcal{T}$ is built from a single convolution with a wider kernel of size $7 \times 7 \times 7$ cleared from any nonlinearities. The minimization problem is carried using an Adam optimizer [49] for 1000 epochs. We use an adaptive step-based scheduler, which decreases the learning rate from $1e^{-4}$ by a factor $\rho = 0.75$ every 250

TABLE III. *A priori* evaluation of the SGS term in developed turbulence regime using testing data with the metrics described in Sec. V A.

	$\downarrow \mathcal{L}_{\text{rms}}(X,Y)$	$\uparrow \mathcal{P}(X,Y)$	$\downarrow \mathcal{J}(P_X\|\|P_Y)$	$\downarrow \mathcal{K}(X,Y)$	$\downarrow \mathcal{I}(X,Y)$
$\mathcal{M}_{\text{DynSmag}}$	0.9400	0.3602	0.2840	0.1742	0.1767
$\mathcal{M}_{\text{DynRG}}$	0.8748	0.4969	0.3471	0.2137	**0.0624**
$\mathcal{M}_{\text{MLP}}$	1.2512	0.1365	**0.0449**	**0.0159**	0.0896
$\mathcal{M}_{\text{CNN}}$	0.8734	0.5597	0.0868	0.0554	0.1021
$\mathcal{M}_{\text{SGTNN}}$	**0.8356**	**0.6118**	0.1070	0.0690	0.1190

epochs. As a preprocessing step, we normalize the input data to zero mean and unitary variance, which has been shown to improve the convergence speed of gradient descent steps [50].

The implementation using PyTorch is available at Ref. [51], and *a priori* results presented below can be reproduced with the given notebooks.

V. RESULTS

We first show in Table II numerical evidence of the unrealistic *a priori* predictions from the $\mathcal{M}_{\text{CNN}}$ w.r.t the considered physical invariances. It is demonstrated that in practice, the $\mathcal{M}_{\text{SGTNN}}$ strictly enforces scalar linearity and Galilean invariance with zero variance over multiple runs of uniformly sampled λ and β coefficients. The soft constraint imposed for rotation invariance reduces the error variance from one order of magnitude over the six tested permutations.

The presented models are now evaluated on different scenarios in two ways: first, we evaluate the performance of a model *a priori* using the metrics described in the previous section, then we run simulations with the surrogate models and evaluate their evolution through time, i.e., *a posteriori*.

A. *A priori* evaluation

1. Developed turbulence regime

We first perform the evaluation on the developed turbulence regime of the testing data set (see Fig. 2), which is the closest to the training data, and where NN models are expected to perform the best. It is important to note that some metrics such as the mean-squared error and correlation coefficient strongly depend on the kernel used to filter out the small scales features, as shown in Ref. [52]. In Table III we show the results of the *a priori* evaluation with the same range of values as the training data, i.e., $\lambda = 1$ and $\beta = 0$. The invariant model $\mathcal{M}_{\text{SGTNN}}$ gives the most consistent structural results compared to the DNS. In particular, we note a substantial improvement on the $\mathcal{L}_{\text{rms}}$ and $\mathcal{P}$ compared to the $\mathcal{M}_{\text{CNN}}$. On the statistical metrics, it is clear from the inset in Fig. 5 that the $\mathcal{M}_{\text{MLP}}$ is better at reproducing mean values (around $\boldsymbol{\nabla} \cdot \mathbf{s} = 0$), which tends to lead to a smaller $\mathcal{J}$ and $\mathcal{K}$. However, the $\mathcal{M}_{\text{SGTNN}}$ is more accurate on the tails of the distribution compared to the other models. This is particularly emphasized by the Quantile-Quantile (QQ) plot, which draws the theoretical quantiles (DNS) on the x-axis and the predicted ones on the y-axis (Fig. 5, right). Again, it is found that the $\mathcal{M}_{\text{SGTNN}}$ is in best agreement with the theoretical quantiles, for which a perfectly reproduced distribution would fit a straight line. During experimentation, we also noticed that even if the best results of $\mathcal{M}_{\text{SGTNN}}$ are achieved with every discussed invariance, the scalar linearity shows the most noticeable impact on *a priori* performance. Since this regime is similar to the training data, models based on machine learning are expected to give optimal results in the term of the target metric $\mathcal{L}$. We see, however, that the $\mathcal{M}_{\text{MLP}}$ is not performing well when evaluated on the divergence. This could be explained by the fact that it is minimizing the error on the residual flux rather than the SGS term itself. The next two regimes will test the ability of the NN-based models to generalize to flows that were not part of the training data.

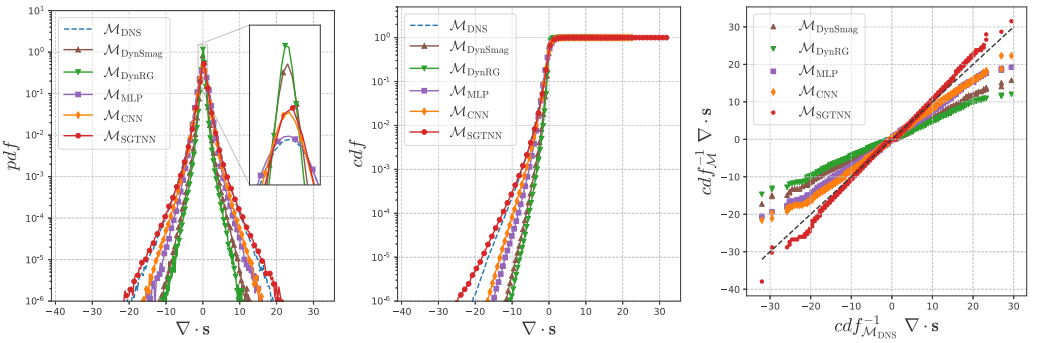

FIG. 5. Probability distribution function (left), cumulative distribution function (middle), and quantiles-quantiles (right) of the SGS term in developed turbulence regime over the entire testing data. Both distribution function tails are made explicitly visible using log scale on the y-axis.

2. Transitional regime

In this regime, we look at a forced flow that transitions from newly initialized to turbulent motion, as seen in Fig. 2. In the *a priori* tests, we consider both velocities and scalar in the transitional regime. Figure 6 shows that the $\mathcal{L}_{\mathrm{rms}}$ is large for the first sample prediction of NN models but rapidly decreases to a similar behavior as in developed turbulence. The correlation given by $\mathcal{P}$ is also quite low during the first iterations but quickly reaches its statistical mean at $t \approx 5$, as well as the metric $\mathcal{J}$ from which a large distance is observed especially from the $\mathcal{M}_{\mathrm{MLP}}$. We can also see in Fig. 7 that the first iterations of the $\mathcal{M}_{\mathrm{MLP}}$ and $\mathcal{M}_{\mathrm{CNN}}$ are producing a non-null SGS term while it is expected to be almost zero, since the presence of turbulence is limited at that time. On the other hand, the $\mathcal{M}_{\mathrm{SGTNN}}$ follows an expected behavior, since it gradually produces a SGS term that

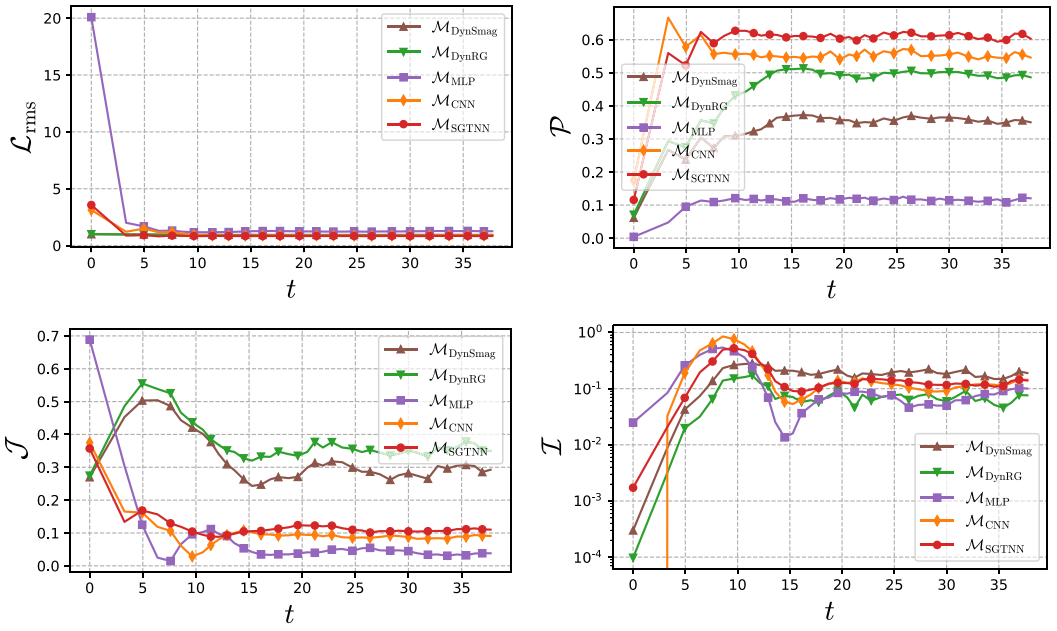

FIG. 6. Time evolution of the different metrics in transitional regime. Normalized root-mean-squared error (top left), Pearson's coefficient (top right), Jensen-Shannon distance (bottom left), and integral dissipation error (bottom right).

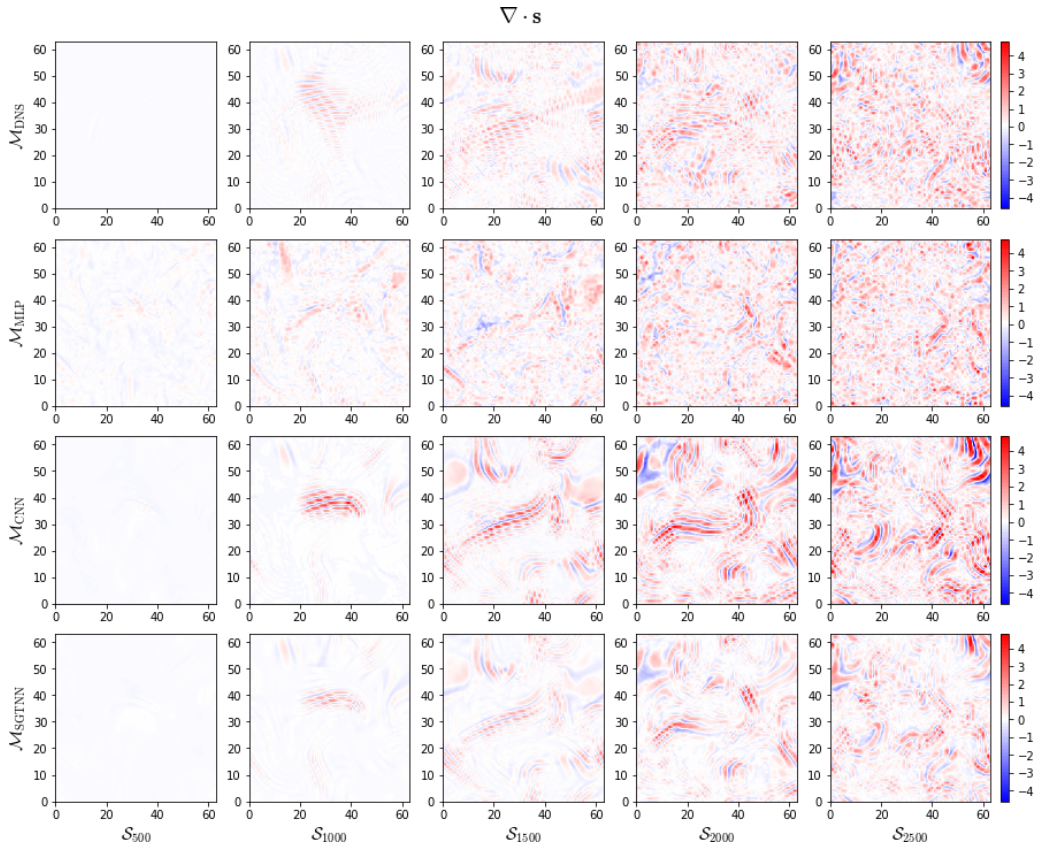

FIG. 7. Data slices of the predicted SGS terms from NN models in transitional regime. It is particularly visible that $\mathcal{M}_{\mathrm{MLP}}$ is producing a non-null SGS term during the first iterations, while $\mathcal{M}_{\mathrm{CNN}}$ quickly produces large localized values.

corresponds to the DNS. A difference that appears for each NN-based model is that they produce a higher dissipation error spike compared the other models. This can be explained by the fact that none of these models have been optimized nor constrained on the scalar variance transfer. Reference [24] also proposes to build a model based on multi-objective minimization, which aims to optimize both functional and structural performances and could also be applied to our model.

3. Scalar decay regime

In this last *a priori* regime, we remove the scalar forcing while maintaining the velocity field in turbulent motion. This results in a decay of the scalar within the domain. This regime is a difficult case for the models presented in this work and assesses their capacity to generalize to flow conditions that have not been seen during the training phase. Both $\mathcal{M}_{\mathrm{MLP}}$ and $\mathcal{M}_{\mathrm{CNN}}$ see their structural and statistical performance drastically decreasing after a short time $t \approx 5$, as seen in Fig. 8 on the $\mathcal{L}_{\mathrm{rms}}$, correlation $\mathcal{P}$, and statistical distance $\mathcal{J}$. Interestingly, we observe that $\mathcal{M}_{\mathrm{CNN}}$ is producing a dissipation error which does not tend toward zero compared to the other models.

Similar to the transitional regime, the data slices in Fig. 9 show clear evidence that the $\mathcal{M}_{\mathrm{MLP}}$ is predicting a large SGS term when the scalar is almost completely mixed. Note that the same behavior is observed for the $\mathcal{M}_{\mathrm{CNN}}$ with a smaller magnitude, but correctly tends to zero for the $\mathcal{M}_{\mathrm{SGTNN}}$ as expected from the DNS.

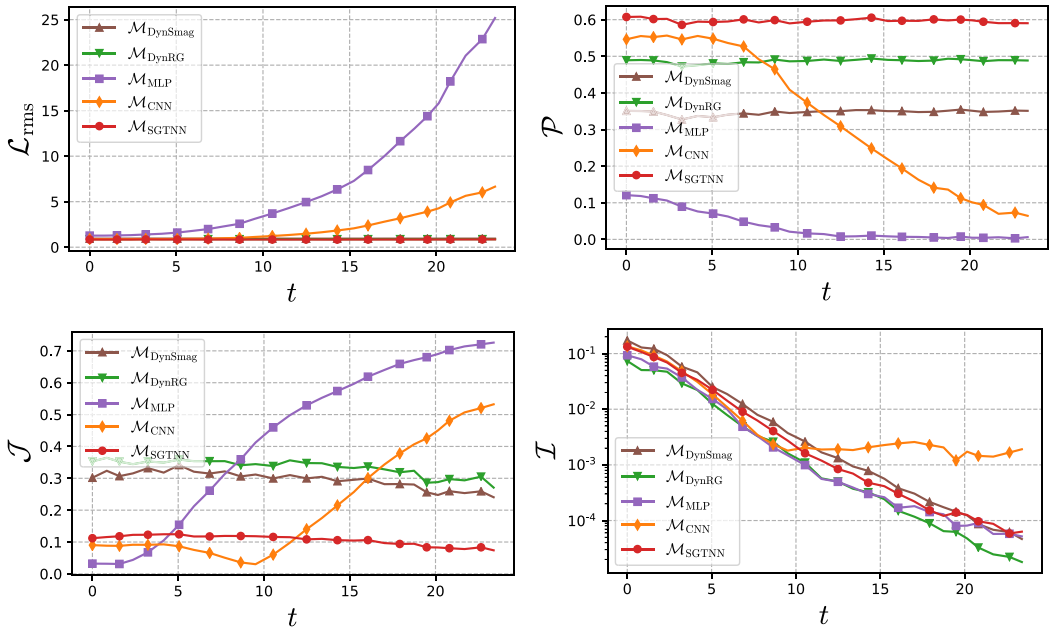

FIG. 8. Time evolution of the different metrics in scalar decay regime. Normalized root-mean-squared error (top left), Pearson's coefficient (top right), Jensen-Shannon distance (bottom left), and integral dissipation error (bottom right).

Overall, when comparing $\mathcal{M}_{\text{CNN}}$ and $\mathcal{M}_{\text{SGTNN}}$, we can say that the imposed invariances act as physical regularizers since they help the generalization while ensuring short-time coherent behavior in different regimes (first iterations in transitional regime and last iterations during scalar decay).

B. *A posteriori* evaluation

We now run new simulations using the NN models to compute the unknown SGS term at every iteration. These *a posteriori* tests are complementary to the *a priori* tests, because they show the behavior and time evolution of the SGS models in practice. In the following tests, we use an hybrid DNS-LES approach in order to isolate the subgrid-scalar modeling from other sources of error. To achieve this, we keep the velocity fields at DNS resolution, i.e., 512^3, while the scalar fields of the different models evolve at the same resolution as the filtered training data, i.e., 64^3. At every substep in the time advancement integration, the velocity fields are extrapolated from the DNS to the LES grid in spectral space. This spectral extrapolation is equivalent to a spectral cutoff filter. Note that only data on LES grid are used to obtain SGS term predictions from algebraic and NN models. The advantage of this procedure is that there is no modeling error on the velocity field used in the scalar equation. Thus, when the scalar LES data are compared with the filtered DNS data, the difference will be due only to the scalar closure term [22,24].

The predictions provided by the different models are able to close the filtered equation (5) such that $\mathcal{M}_{\text{NN}}(\overline{\mathbf{u}}, \overline{\Phi}) = \boldsymbol{\nabla} \cdot \mathbf{s}$ given the low-resolution scalar and filtered velocities. Now we look at different statistical metrics such as the filtered scalar variance $\langle \overline{\Phi}^2 \rangle - \langle \overline{\Phi} \rangle^2$ and the resolved scalar enstrophy $\langle \overline{\omega}^2 \rangle / 2 = \langle \nabla^2 \overline{\Phi} \rangle$, which is particularly interesting to characterizes the energy transfers in the smallest resolved scales [53], from which the SGS model is the most important. For each regime, we also plot the scalar energy spectrum computed using the different SGS models at intermediate and final time in the simulation.

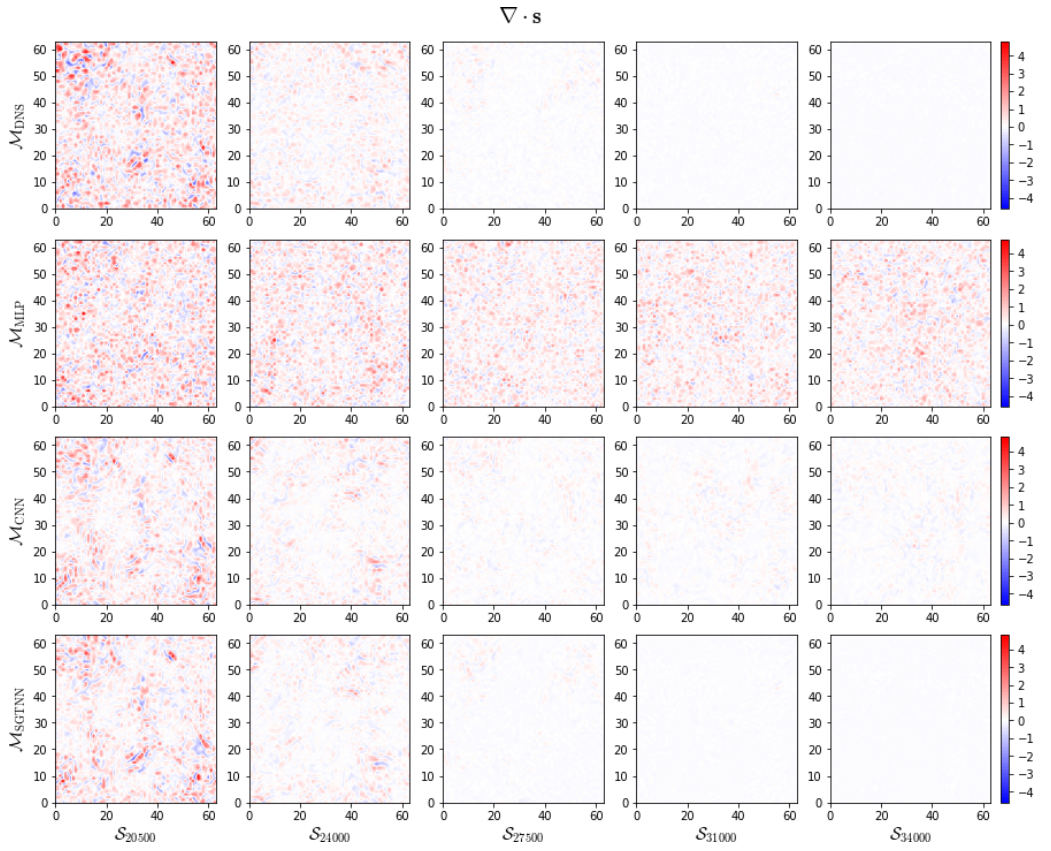

FIG. 9. Data slices of the predicted SGS terms from NN models in scalar decay regime. It is particularly visible that $\mathcal{M}_{\mathrm{MLP}}$ and $\mathcal{M}_{\mathrm{CNN}}$ are still predicting a non-null SGS term when the scalar is completely mixed.

1. Developed turbulence regime

In the first two simulations, we start in a developed turbulence regime at $\mathcal{S} = 20\,000$ of the training data (see Fig. 1) and the testing data (see Fig. 2), respectively, with the same velocity and

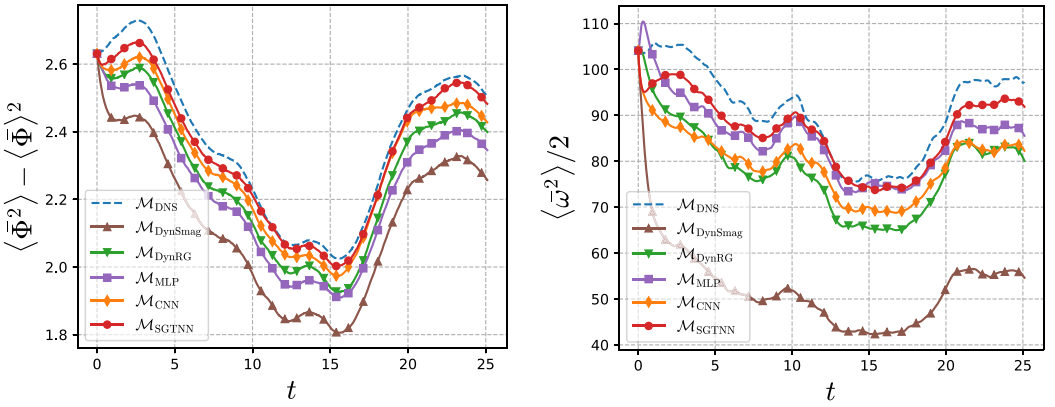

FIG. 10. Statistics of simulation in developed turbulence regime starting from training data. Scalar variance (left), resolved scalar enstrophy (right).

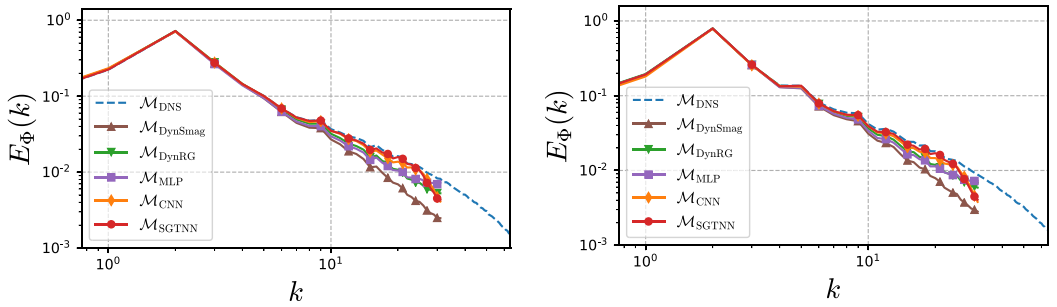

FIG. 11. Energy spectrum of simulation in developed turbulence regime starting from training data at $t = 12.5$ (left) and $t = 25$ (right).

scalar forcing as used in the data sets. Using the same starting point as the training data is useful in order to consider the accumulation of errors during the simulation, since the models learnt from only a finite number of samples of the exact simulation. We see in Fig. 10 that the scalar variance, on the left, produced by the $\mathcal{M}_{\text{SGTNN}}$ and the $\mathcal{M}_{\text{CNN}}$ are the closest from the DNS except after a long integration time, $t \approx 20$ where the $\mathcal{M}_{\text{CNN}}$ starts accumulating error. The behavior of the model on the largest scales in shown by the resolved scalar enstrophy on the right. Here the $\mathcal{M}_{\text{MLP}}$ has a consistent behavior almost similar to the $\mathcal{M}_{\text{SGTNN}}$. One can verify in the spectra from Fig. 11 that the two models presented in this work are the most accurate on the smallest wave numbers, while the NN baseline model from Ref. [25] is most effective close to the spectral cut at $k \geqslant 30$. Now, the same conclusions can be drawn with the testing data starting point as depicted in Figs. 12 and 13. We can already see that a CNN without physical constraints is not able to reproduce the dynamics of the SGS term in statistically similar conditions as those of the training data. Indeed, it performs at the same accuracy for the scalar variance but worse for the resolved scalar enstrophy than some algebraic models. The effect of the physical invariances is primordial and gives the ability to get the best performances from the neural network.

2. Scalar transition regime

The scalar transition regime is already testing the extrapolation capabilities of the model. In this regime, we start from a velocity field already in turbulent motion and a scalar initialized at large scales and forced with the previously used scheme. The difficulty of this regime resides in the transition of the scalar field to turbulent advection, which then comes down to the previous regime

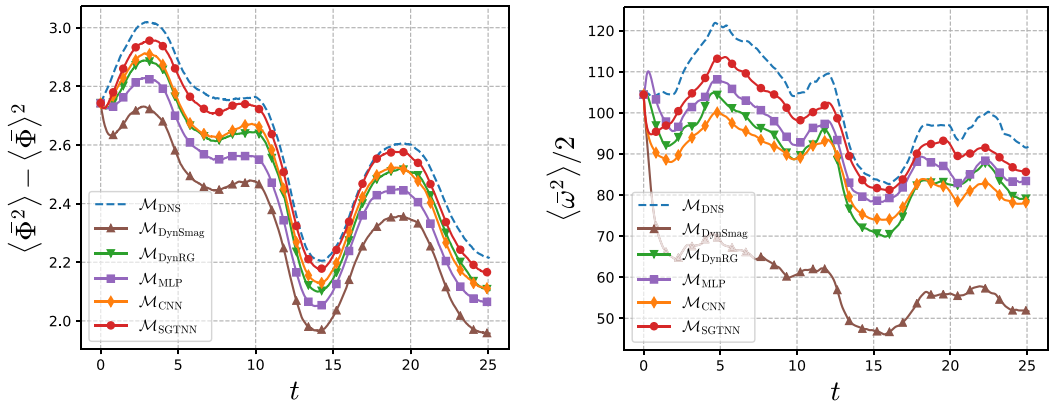

FIG. 12. Statistics of simulation in developed turbulence regime starting from testing data. Scalar variance (left), resolved scalar enstrophy (right).

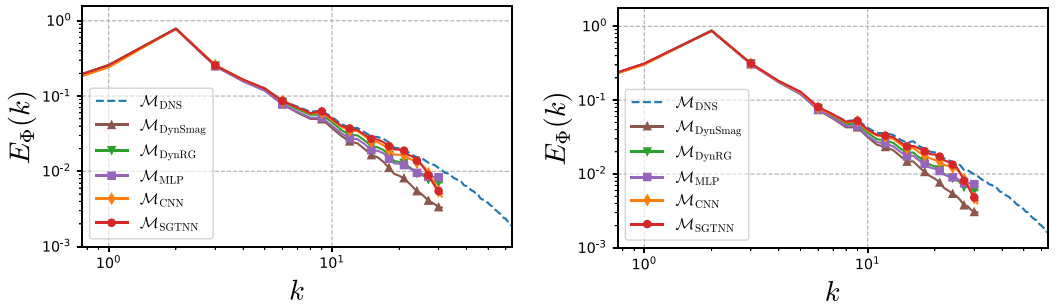

FIG. 13. Energy spectrum of simulation in developed turbulence regime starting from testing data at $t = 12.5$ (left) and $t = 25$ (right).

after $t = 15$. Both scalar variance and resolved enstrophy are very well reproduced by the $\mathcal{M}_{\text{SGTNN}}$ (see Fig. 14), which gives superior performances compared to the algebraic models. The hypothesis drawn from the *a priori* test of the similar regime seems to correlate with the *a posteriori* statistics, where both $\mathcal{M}_{\text{CNN}}$ and $\mathcal{M}_{\text{MLP}}$ produce a large error during the first transitional iterations, while the invariant NN model is more consistent. However, Fig. 15 indicates that the spectra produced by the CNN models are closer to those of the DNS compared to the baseline models. In particular, we can see on the left that most significant deviation appears during the transition to turbulent advection, i.e., the first iterations of the simulation.

3. *Full decay regime*

We saw in *a priori* tests that the $\mathcal{M}_{\text{SGTNN}}$ gave the best performances in the scalar decay regime while the other NN models were not able to generalize. In *a posteriori*, we go one step further and completely remove the source terms for both velocity and scalar fields, which results in a full decay. However, we still observe the same behavior in Fig. 16. More precisely, the resolved scalar enstrophy shows that the $\mathcal{M}_{\text{MLP}}$ and $\mathcal{M}_{\text{CNN}}$ are still producing energy transfers after a long time. We also see in the spectrum (Fig. 17) that both unconstrained NN models are not able to dissipate the large scales of the SGS term at $t = 8$, which results in a blowup of the simulation after $t = 30$. The $\mathcal{M}_{\text{SGTNN}}$, however, is still in excellent agreement with the DNS and outperforms significantly the algebraic models, even in regimes that were not part of the training data.

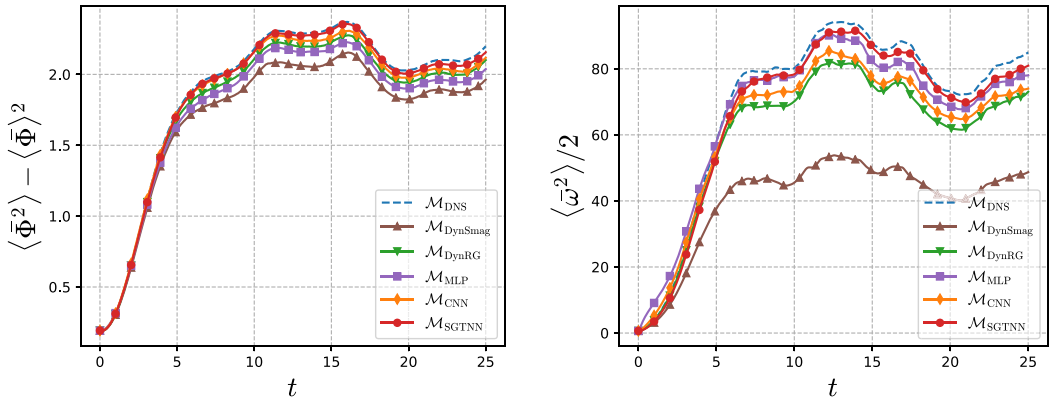

FIG. 14. Statistics of simulation in scalar transition regime, starting from a newly initialized field. Scalar variance (left), resolved scalar enstrophy (right).

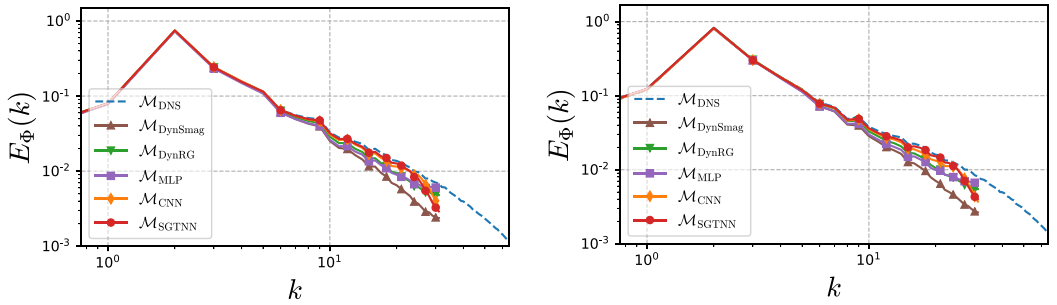

FIG. 15. Energy spectrum of simulation in scalar transition regime, starting from a newly initialized field at $t = 6$ (left) and $t = 25$ (right).

4. Generalization capabilities

The *a posteriori* regimes discussed above are statistically different from the data used to train the NN models. To evaluate the robustness of the models and their ability to extrapolate to different regimes, we compute the largest absolute error of the previously shown flow statistics, namely, scalar variance and enstrophy during their time evolution,

$$\mathcal{L}_{\max}(X, Y) = \max |X - Y|/Y, \tag{20}$$

where Y and X refer to the flow statistics computed from DNS simulations and from LES simulations with the SGS model, respectively. The results are shown in Table IV for the scalar variance and resolved scalar enstrophy. For each regime, we also compare the current $\mathcal{L}_{\max}(X, Y)$ with the reference (or base) given by the developed turbulence regime starting from training data. For the scalar variance, the results given by the $\mathcal{M}_{\mathrm{MLP}}$ are quite robust in the different regimes, and the maximum error in full decay is only $\times 1.8275$ higher compared to the base regime. The $\mathcal{M}_{\mathrm{CNN}}$ is consistent in the first regimes, but its performance dramatically drops in the full decay. The $\mathcal{M}_{\mathrm{SGTNN}}$, however, is able to maintain a stable performance with an error in full decay only $\times 1.1509$ higher than in developed turbulence. For the resolved scalar enstrophy, we can see that the $\mathcal{M}_{\mathrm{MLP}}$ is not even robust to the scalar transition, with an error more than $\times 20$ higher than in the base regime. The $\mathcal{M}_{\mathrm{CNN}}$ sees its performance gradually degraded throughout the regimes of increasing difficulty. The same comments are also valid for the $\mathcal{M}_{\mathrm{SGTNN}}$, except that the maximum error has increased

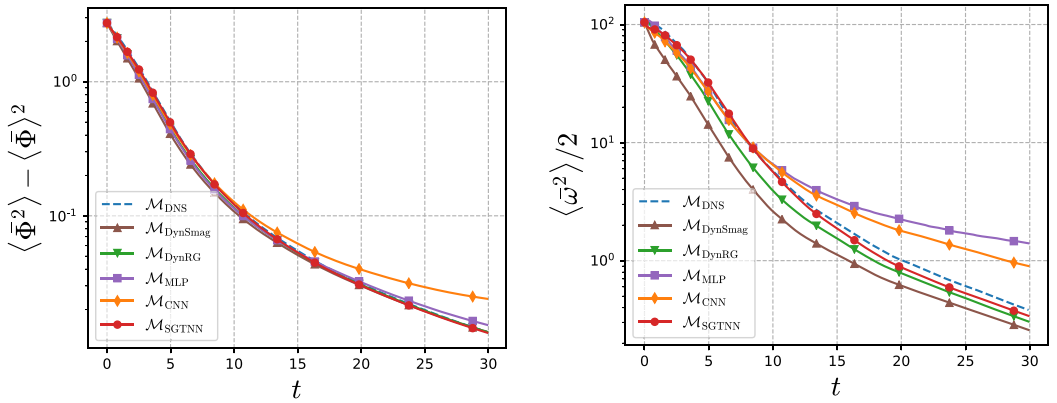

FIG. 16. Statistics of simulation in full decay regime starting from testing data. Scalar variance (left), resolved scalar enstrophy (right).

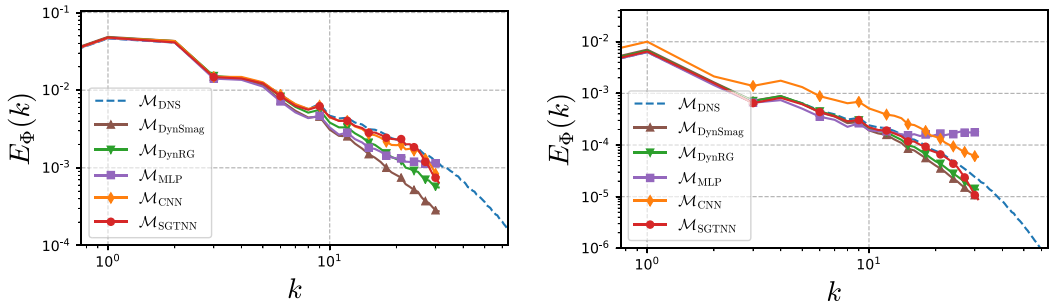

FIG. 17. Energy spectrum of simulation in full decay regime starting from testing data at $t = 8$ (left) and $t = 30$ (right).

by a factor of $\times 1.6841$ in full decay, which is five times smaller than the error increase of the model without physical invariances.

VI. SUMMARY

We propose a closure model to the problem of scalar transport in a turbulent incompressible flow based on NNs and physical invariances. The model is shown to predict a realistic behavior and outperforms classical algebraic models on different metrics. During the *a posteriori* evaluation, we demonstrate the capabilities of the model to generalize to unseen flow regimes compared to NN models that do not embed physical knowledge. Thus, we think that to be applicable to large-scale simulations, NN models will require some form of physical invariance. Other interesting directions towards a generic model would include an explicit mention of the filter width, Reynolds number, and Schmidt number. Learning a dynamic coefficient related to the filter width could be the subject of future investigations. The ideal setting of spectral discretization also introduces some limitations with respect to the geometry of the domain. Moving to classical numerical methods such as finite volumes or finite differences constitutes a next exploratory step that comes with difficulties related to the separation of physical quantities and discretization errors introduced by the chosen scheme.

TABLE IV. Maximum error $\mathcal{L}_{\max}$ on scalar variance (first three rows) and resolved scalar enstrophy (last three rows) of the different NN models during each *a posteriori* regime. Relative error compared to the base regime, i.e., developed turbulence starting from training data, is shown as $\mathcal{L}_{\max}$/ base.

	$\mathcal{L}_{\max}$ (base)	$\mathcal{L}_{\max}$	$\mathcal{L}_{\max}$	$\mathcal{L}_{\max}$
$X = \langle\bar{\Phi}^2\rangle - \langle\bar{\Phi}\rangle^2$:				
$\mathcal{M}_{\mathrm{MLP}}$	0.0719 ($\times$1)	0.0839 ($\times$1.1669)	0.0731 ($\times$1.0167)	0.1314 ($\times$1.8275)
$\mathcal{M}_{\mathrm{CNN}}$	0.0405 ($\times$1)	0.0578 ($\times$1.4272)	0.0389 ($\times$0.9605)	0.7650 ($\times$18.8889)
$\mathcal{M}_{\mathrm{SGTNN}}$	**0.0265** ($\times$1)	**0.0268** (**$\times$1.0113**)	**0.0193** (**$\times$0.7283**)	**0.0305** (**$\times$1.1509**)
$X = \langle\bar{\omega}^2\rangle/2$:				
$\mathcal{M}_{\mathrm{MLP}}$	0.1180 ($\times$1)	0.1271 (**$\times$1.0771**)	2.6277 ($\times$22.2686)	2.6676 ($\times$22.6068)
$\mathcal{M}_{\mathrm{CNN}}$	0.1747 ($\times$1)	0.1890 ($\times$1.0818)	0.4938 ($\times$2.8266)	1.3529 ($\times$7.7441)
$\mathcal{M}_{\mathrm{SGTNN}}$	**0.0823** ($\times$1)	**0.0908** ($\times$1.1033)	**0.1117** (**$\times$1.3572**)	**0.1386** (**$\times$1.6841**)
	Developed (train)	Developed (test)	Scalar transition	Full decay

ACKNOWLEDGMENTS

This work was supported by the CNRS through the 80 PRIME project and the ANR through the Melody, OceaniX, and HRMES projects. Computations were performed using HPC and GPU resources from GENCI-IDRIS (Grants 020611 and 101030) and GRICAD infrastructures.

[1] K. Duraisamy, G. Iaccarino, and H. Xiao, Turbulence modeling in the age of data, Annu. Rev. Fluid Mech. **51**, 357 (2019).

[2] R. Wang, K. Kashinath, M. Mustafa, A. Albert, and R. Yu, Towards physics-informed deep learning for turbulent flow prediction, in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (ACM, Virtual Event, CA, 2020), pp. 1457–1466.

[3] G. D. Portwood, P. P. Mitra, M. D. Ribeiro, T. M. Nguyen, B. T. Nadiga, J. A. Saenz, M. Chertkov, A. Garg, A. Anandkumar, A. Dengel, R. Baraniuk, and D. Schmidt, Turbulence forecasting via neural ODE, presented at Machine Learning and the Physical Sciences Workshop at the 33rd Conference on Neural Information Processing Systems, Vancouver, CA, December 2019.

[4] A. T. Mohan, N. Lubbers, D. Livescu, and M. Chertkov, Embedding hard physical constraints in convolutional neural networks for 3D turbulence, in *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations* (ICLR, Virtual Event, Addis Ababa Ethiopia, 2020).

[5] M. Gamahara and Y. Hattori, Searching for turbulence models by artificial neural network, Phys. Rev. Fluids **2**, 054604 (2017).

[6] C. Xie, J. Wang, and E. Weinan, Modeling subgrid-scale forces by spatial artificial neural networks in large eddy simulation of turbulence, Phys. Rev. Fluids **5**, 054606 (2020).

[7] R. Maulik, O. San, A. Rasheed, and P. Vedula, Subgrid modeling for two-dimensional turbulence using neural networks, J. Fluid Mech. **858**, 122 (2019).

[8] C. Xie, J. Wang, K. Li, and C. Ma, Artificial neural network approach to large-eddy simulation of compressible isotropic turbulence, Phys. Rev. E **99**, 053113 (2019).

[9] A. Beck, D. Flad, and C.-D. Munz, Deep neural networks for data-driven LES closure models, J. Comput. Phys. **398**, 108910 (2019).

[10] S. Pawar, O. San, A. Rasheed, and P. Vedula, A priori analysis on deep learning of subgrid-scale parametrizations for Kraichnan turbulence, Theor. Comput. Fluid Dyn., **34**, 429 (2020).

[11] C. J. Lapeyre, A. Misdariis, N. Cazard, D. Veynante, and T. Poinsot, Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates, Combust. Flame **203**, 255 (2019).

[12] T. Bolton and L. Zanna, Applications of deep learning to ocean data inference and subgrid parameterization, J. Adv. Model. Earth Syst. **11**, 376 (2019).

[13] J. Ling, A. Kurzawski, and J. Templeton, Reynolds averaged turbulence modeling using deep neural networks with embedded invariance, J. Fluid Mech. **807**, 155 (2016).

[14] J. Ling, R. Jones, and J. Templeton, Machine learning strategies for systems with invariance properties, J. Comput. Phys. **318**, 22 (2016).

[15] M. Bode, M. Gauding, Z. Lian, D. Denker, M. Davidovic, K. Kleinheinz, J. Jitsev, and H. Pitsch, Using physics-informed enhanced super-resolution generative adversarial networks for subfilter modeling in turbulent reactive flows, Proc. Combust. Inst., in press, corrected proof, available 25 January 2021.

[16] P. Sagaut, *Large Eddy Simulation for Incompressible Flows: An Introduction* (Springer-Verlag, Berlin Heidelberg, 2006).

[17] J. Smagorinsky, General circulation experiments with the primitive equations: I. The basic experiment, Mon. Weather Rev. **91**, 99 (1963).

[18] R. A. Clark, J. H. Ferziger, and W. C. Reynolds, Evaluation of subgrid-scale models using an accurately simulated turbulent flow, J. Fluid Mech. **91**, 1 (1979).

[19] J. Bardina, J. Ferziger, and W. Reynolds, Improved subgrid-scale models for large-eddy simulation, in *13th Fluid and Plasmadynamics Conference* (American Institute of Aeronautics and Astronautics, Snowmass, CO USA, 1980), p. 1357.

[20] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot, A dynamic subgrid-scale eddy viscosity model, Phys. Fluids A **3**, 1760 (1991).
[21] D. K. Lilly, A proposed modification of the Germano subgrid-scale closure method, Phys. Fluids A **4**, 633 (1992).
[22] G. Balarac, J. Le Sommer, X. Meunier, and A. Vollant, A dynamic regularized gradient model of the subgrid-scale scalar flux for large eddy simulations, Phys. Fluids **25**, 075107 (2013).
[23] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature (London) **521**, 436 (2015).
[24] A. Vollant, G. Balarac, and C. Corre, Subgrid-scale scalar flux modeling based on optimal estimation theory and machine-learning procedures, J. Turbul. **18**, 854 (2017).
[25] G. D. Portwood, B. T. Nadiga, J. A. Saenz, and D. Livescu, Interpreting neural network models of residual scalar flux, J. Fluid Mech. A **907**, 23 (2021).
[26] G. K. Batchelor, Small-scale variation of convected quantities like temperature in turbulent fluid Part 1. General discussion and the case of small conductivity, J. Fluid Mech. **5**, 113 (1959).
[27] A. M. Obukhov, Structure of temperature field in turbulent flow, Izvestiia Akademii Nauk SSSR, Ser. Geogr. i Geofiz. **13**, 58 (1970).
[28] S. Corrsin, On the spectrum of isotropic temperature fluctuations in an isotropic turbulence, J. Appl. Phys. **22**, 469 (1951).
[29] C. Rosales and C. Meneveau, Linear forcing in numerical simulations of isotropic turbulence: Physical space implementations and convergence properties, Phys. Fluids **17**, 095106 (2005).
[30] V. Eswaran and S. B. Pope, An examination of forcing in direct numerical simulations of turbulence, Comput. Fluids **16**, 257 (1988).
[31] K. Alvelius, Random forcing of three-dimensional homogeneous turbulence, Phys. Fluids **11**, 1880 (1999).
[32] C. B. da Silva and J. C. Pereira, Analysis of the gradient-diffusion hypothesis in large-eddy simulations based on transport equations, Phys. Fluids **19**, 035106 (2007).
[33] A. Gorbunova, G. Balarac, M. Bourgoin, L. Canet, N. Mordant, and V. Rossetto, Analysis of the dissipative range of the energy spectrum in grid turbulence and in direct numerical simulations, Phys. Rev. Fluids **5**, 044604 (2020).
[34] J.-B. Lagaert, G. Balarac, and G.-H. Cottet, Hybrid spectral-particle method for the turbulent transport of a passive scalar, J. Comput. Phys. **260**, 127 (2014).
[35] G. Balarac, H. Pitsch, and V. Raman, Development of a dynamic model for the subfilter scalar variance using the concept of optimal estimators, Phys. Fluids **20**, 035114 (2008).
[36] P. Moin, K. Squires, W. Cabot, and S. Lee, A dynamic subgrid-scale model for compressible turbulence and scalar transport, Phys. Fluids A **3**, 2746 (1991).
[37] A. Leonard, Energy cascade in large-eddy simulations of turbulent fluid flows, Adv. Geophys. **18**, 237 (1974).
[38] C. Meneveau and J. Katz, Scale-invariance and turbulence models for large-eddy simulation, Annu. Rev. Fluid Mech. **32**, 1 (2000).
[39] D. M. Endres and J. E. Schindelin, A new metric for probability distributions, IEEE Trans. Inf. Theory **49**, 1858 (2003).
[40] J. Hodges, The significance probability of the Smirnov two-sample test, Ark. Mat. **3**, 469 (1958).
[41] L. C. Berselli, T. Iliescu, and W. J. Layton, *Mathematics of Large Eddy Simulation of Turbulent Flows* (Springer-Verlag, Berlin Heidelberg, 2006).
[42] C. G. Speziale, Galilean invariance of subgrid-scale stress models in the large-eddy simulation of turbulence, J. Fluid Mech. **156**, 55 (1985).
[43] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, Real-time motor fault detection by 1-D convolutional neural networks, IEEE Trans. Ind. Elect. **63**, 7067 (2016).
[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Imagenet Classification with Deep Convolutional Neural Networks, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., Lake Tahoe, Nevada USA, 2012), pp. 1097–1105.

[45] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker, Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation, Med. Image Anal. **36**, 61 (2017).
[46] M. Oberlack, Invariant modeling in large-eddy simulation of turbulence, Ann. Res. Briefs, 3 (1997).
[47] T. Cohen and M. Welling, Group Equivariant Convolutional Networks, in *ICML'16: Proceedings of the 33nd International Conference on Machine Learning*, edited by M. F Balcan and K. Q. Weinberger (PMLR, New York City, NY, 2016), Vol. 48, pp. 2990–2999.
[48] M. Weiler, M. Geiger, M. Welling, W. Boomsma, and T. S. Cohen, 3D steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., Palais des Congrès de Montréal, Montréal Canada, 2018), pp. 10381–10392.
[49] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, in *ICLR 2015 3rd International Conference on Learning Representations* (2015).
[50] S. Ioffe and C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in *ICML'15: Proceedings of the 32nd International Conference on Machine Learning*, edited by B. Francis and B. David (PMLR, Lille, France, 2015), Vol. 37, pp. 448–456.
[51] H. Frezat, G. Balarac, J. Le Sommer, R. Fablet, and R. Lguensat, Subgrid Transport Neural Network (2020), https://github.com/hrkz/subgridtransportNN.
[52] Y. Fabre and G. Balarac, Development of a new dynamic procedure for the Clark model of the subgrid-scale scalar flux using the concept of optimal estimator, Phys. Fluids **23**, 115103 (2011).
[53] O. San, A. E. Staples, and T. Iliescu, A posteriori analysis of low-pass spatial filters for approximate deconvolution large eddy simulations of homogeneous incompressible flows, Int. J. Comput. Fluid Dyn. **29**, 40 (2015).

Correction: References [3] and [15] were updated incorrectly at the proof stage and have been fixed. A reference citation below Eq. (14) has been fixed.