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

(Dated: October 12, 2020)
Abstract

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 invariances and symmetries derived from physical laws. From simulation-based experiments, we show that the proposed physically-invariant NN model outperforms both purely data-driven ones as well as parametric state-of-the-art subgrid-scale model. 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 physically-invariant NN is shown to generalize to configurations that have not been seen during the training phase.

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 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 (NN) 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 (CNN) \[9, 10\]. 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 parameterization 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 only able to predict 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 on different configurations of 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. Sec. 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 in three different configurations in Sec. V.

II. SGS SCALAR MODELING

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

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

with velocity \( \mathbf{u} = (u_x, u_y, u_z) \), density \( \rho \), pressure \( p \), kinematic viscosity \( \nu \) and \( \mathbf{F_u} \) an external forcing. We define the advection-diffusion of a scalar \( \Phi \) defined by its Schmidt number and
molecular diffusivity \( \kappa = \nu/Sc \)

\[
\frac{\partial \Phi}{\partial t} + (u \cdot \nabla)\Phi = \nabla \cdot (\kappa \nabla \Phi)
\] (2)

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

\[
\Phi(x) = \int_V \Phi(x_f) G(x_f - x) \, dx_f
\] (3)

where the kernel \( G \) is concentrated in a band defined by the filter width. Applying the filter (3) on (2) yields the filtered advection-diffusion equation

\[
\frac{\partial \Phi}{\partial t} + (\bar{u} \cdot \nabla)\Phi = \nabla \cdot (\kappa \nabla \Phi) + \nabla \cdot (\bar{u} \Phi - \bar{u} \Phi)
\] (4)

in which the residual flux is defined exactly as,

\[
s \equiv \bar{u} \Phi - \bar{u} \Phi
\] (5)

Thus, the unknown subgrid-scale (SGS) term that appears in (4) is the divergence of the residual flux, i.e. \( \nabla \cdot s \). 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 \textit{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 \textit{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 configurations, 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 [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 model requires the generation of a training dataset, usually coming from DNS, from which model parameters are learnt. This dataset 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 dataset 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

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

(6)

where $C_\Phi$ is the Oboukou-Corrsin constant, $\chi$ and $\epsilon$ are the scalar-variance and kinetic-energy dissipation rate respectively in spectral space at wavenumber $k$. 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 configuration.

III.1. Data generation

Numerical solutions of the equations (1) and (2) are computed using a classical Fourier pseudo-spectral solver in a triply periodic domain $[0, 2\pi)^3$ and using a random forcing [29]. This allows us to remove any spatial discretization error and focus on the modeling of the SGS residual flux divergence. Thus, we extract filtered SGS term $\mathcal{M}_{DNS}$ and non-residual inputs quantities that appears in (4), i.e

$$\mathcal{M}_{DNS}(i) = \nabla \cdot s, \quad i = \{\mathbf{u}, \Phi\}$$

(7)

Reference data are then obtained from DNS using $N = 512^3$ grid points at Reynolds number $Re_\lambda = 160$ and Schmidt number $Sc = 0.7$. The scalar is modulated by a spectral forcing that applies on the smallest wavenumbers in order to contain its effect to the resolved scales of the simulation. To avoid large correlations in the data, we ensure that every sample is
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 $S = 20000$. The first 50 samples are used for training and the remaining 40 samples are used for validation.

FIG. 2. Evolution of the Reynolds number on the Taylor microscale (left) and scalar variance (right) in the different regimes. The same flow configuration with a new initialization has been used for the testing data.

separated by almost one large eddy turnover time $t_L \sim L/U \sim (L^2/\epsilon)^{1/3}$ from previous and following ones and construct a dataset from 80 equally-spaced samples taken from 60000 iterations. The dataset 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 dataset, we follow the same methodology and extract three different flow regimes, i.e. developed turbulence, transitional and scalar decay. Reynolds number and scalar variance evolution as well as data separation from the datasets 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 $S$ taken from the
FIG. 3. Data slices extracted during the simulation. $\mathbf{u}$ are the velocities in each directions, $\Phi$ is the advected scalar and $\mathcal{M}_{\text{DNS}}$ 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 $S = 20000$. 

training dataset.
III.2. Baseline models

For benchmarking purposes, we consider two algebraic closure models and a recent model based on machine learning. Both algebraic models are based on a dynamic 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 [30] is an extension of the Smagorinsky model [20] for scalar transport. This is a functional model based on an eddy diffusivity leading to

\[
s_{\text{DynSMAG}} = C \hat{\Delta}^2 ||\bar{S}|| \frac{\partial \Phi}{\partial x_j} \tag{8}
\]

where \( ||\bar{S}|| \) is the \( L_2 \) norm of the resolved strain rate tensor. This model is commonly used because it is stable under gradient direction, i.e. it always produces forward-scatter when \( C \in \mathbb{R}_+^* \).

The gradient model on the other side is a structural model based on a truncated series expansion using the filter scale [31]. This model is known to provide a good local approximation and correlation with the exact residual flux. 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 to a regularization of the model that eliminates back-scatter transfers [22]. The dynamic regularized gradient model then writes

\[
s_{\text{DynRG}} = C \hat{\Delta}^2 S_{ij}^\ominus \frac{\partial \Phi}{\partial x_j} \tag{9}
\]

where \( S_{ij}^\ominus \) only contains 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, \( \bar{u} \) and \( \bar{\Phi} \) as inputs and optimizes \( L \) fully connected linear (or affine) layers with non-linear ReLu activations \( R(x) = \max(0, x) \) such that

\[
s_{\text{MLP}} = W^{(L)} h^{(L-1)} + b^{(L)}, \quad h^{(i)} = R(W^{(i)} h^{(i-1)} + b^{(i)}), \quad h^{(0)} = \{ \bar{u}, \bar{\Phi} \} \tag{10}
\]

where \( b \) is a bias vector and \( W \) a weight matrix. To our knowledge, this model is the closest to our work.
III.3. 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 $s \cdot \nabla \Phi$. In this paper however, we focus on the modeling of the SGS term $\nabla \cdot s$ and thus our proposed model does not have access to the residual flux $s$. Therefore, we perform an extensive study of the predicted SGS term (or structural performance) statistical properties. As discussed in [32], 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. The structural evaluation is given by the mean-squared-error (or $L_2$) and Pearson’s cross-correlation coefficient $P$ between the predicted SGS term and the filtered DNS. We also show the error on the integral of the SGS dissipation $I$, which can be computed using the divergence of the SGS flux since

$$\int_V s \cdot \nabla \Phi \, dV = - \int_V \Phi \nabla \cdot s \, dV$$

(11)

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 $J$ [33] 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-Leiber divergence $D$, or relative entropy. Besides, we compute a Kolmogorov-Smirnov statistic test $K$ that hypothesizes that two samples are drawn from the same distributions if its value is small ($\leq 0.1$) as shown by [34].

IV. PHYSICALLY-INVARIANT LEARNING FRAMEWORK

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

$$M_{\text{NN}}(\mathbf{u}, \Phi) \approx \nabla \cdot s$$

(12)
We choose four non-exhaustive 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 [35] 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.

IV.1. 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

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

where $T_\delta$ is a translation operator for spatial displacement $\delta$. In the deep learning literature, in contrast to fully-connected architectures [36] used in [3], Convolutional Neural Networks (CNNs) are described by a combination of convolution and activation layers. Since the convolution commutes with translations, this type of architecture provides a built-in invariance to spatial displacements, which satisfy (13). We note that using a different set of inputs can already ensure some symmetries of the residual flux [37] but reduces the flexibility of the model.

| Structural metric | $L_2(X, Y) = \frac{1}{N}||X - Y||_2$ |
|-------------------|--------------------------------------|
| Pearson’s coefficient | $\mathcal{P}(X,Y) = \text{cov}(X,Y)/\sigma_X\sigma_Y$ |

| Physical metric | $\mathcal{I}(X,Y) = -\int_V \nabla \Phi \cdot \mathbf{s} \, dV + \int_V \nabla \Phi \cdot \Phi \, dV$ |

| Statistical metric | $\mathcal{D}(P_X||P_Y) = \sum_i^N P_X(i) \log (P_X(i)/P_Y(i))$ |
|-------------------|-------------------------------------------------|
| Jensen-Shannon distance | $\mathcal{J}(P_X||P_Y) = \sqrt{\frac{1}{2}\mathcal{D}(P_X||M) + \frac{1}{2}\mathcal{D}(P_Y||M)}, \quad M = \frac{1}{2}(P_X + P_Y)$ |
| Kolmogorov-Smirnov statistic | $\mathcal{K}(X,Y) = \sup_t |F_X(t) - F_Y(t)|, \quad F_A(t) = \frac{1}{N} \sum_i^N 1_{A_i \leq t}$ |

TABLE I. Equations for the structural, physical and statistical metrics used to evaluate the a priori performance of the different models.
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 [38], two-dimensional [39] or three-dimensional [40] states.

Let us call now $\mathcal{M}_{\text{CNN}}$ a model built from convolution and non-linear 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 physically-invariant NN model as $\mathcal{M}_{\text{SGTNN}}$ for "Subgrid Transport Neural Network".

### IV.2. 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

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

This relationship can not be verified within $\mathcal{M}_{\text{CNN}}$ since the model contains non-linearities introduced by the $\mathcal{R}$ operator. We propose to split the model $\mathcal{M}_{\text{CNN}}$ into two sub-models $\mathcal{V}$ 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 non-linear activation. The resulting SGS model is given by

$$\mathcal{M}_{\text{SGTNN}}(\mathbf{u}, \Phi) = [\mathcal{V}(\mathbf{u}) \times \mathcal{T}(\Phi)] * \mathcal{C}^+ \quad (15)$$

where $*$ is the discrete convolution operator, $\times$ 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-
FIG. 4. Illustration of the subgrid transport neural network architecture (SGTNN). On the left, 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 ensure Galilean invariance. Then, the two sub-models $\mathcal{V}$ for the velocity and $\mathcal{T}$ for the scalar transport are represented in dotted and dashed lines respectively. The convolution with kernel size $K = 1$ corresponding to $C^+$ applied to the combined result of the latter is depicted on the right. The 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 show that the relationship (14) holds,

$$M_{\text{SGTNN}}(\bar{u}, \lambda \Phi) = \lambda \left( \sum_{i=0}^{d} \Phi_i \right) = \lambda \left( \sum_{i=0}^{d} \Phi_i \right)$$

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

$$M_{\text{SGTNN}}(\bar{u}, \lambda \Phi) = \lambda \left( \sum_{i=0}^{d} \Phi_i \right)$$

which validates that the architecture $M_{\text{SGTNN}}$ has a linear path on $\Phi$.

IV.3. 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 [37]. 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

\[ \forall \beta \in \mathbb{R}, \quad M_{\text{NN}}(\mathbf{u} + \beta, \Phi) = M_{\text{NN}}(\mathbf{u}, \Phi) \quad (16) \]

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

\[ (\mathbf{u} + \beta)' = \mathbf{u} + \beta - \langle \mathbf{u} + \beta \rangle \]
\[ = \mathbf{u} - \langle \mathbf{u} \rangle \]
\[ = (\bar{\mathbf{u}})' \]

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

**IV.4. Rotation invariance**

The last fundamental invariance required to fulfil frame symmetry is related to reflections and rotations \[41\]. The filtered equation (4) holds under the rotation of the coordinate system and velocity vector,

\[ M_{\text{NN}}(A_{ij} \bar{\mathbf{u}}, \Phi) = M_{\text{NN}}(\mathbf{u}, \Phi) \quad (17) \]

for any rotation matrix \( A \) with \( A^T A = A A^T = I \) in a rotated frame \( \mathbf{x}_i = A_{ij} \mathbf{x}_j \). 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 \[42, 43\] 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 dataset with new samples that exhibit the desired invariance. We used permutations of our initial data that verifies (17) with angles of 90 degrees.

**IV.5. NN architecture and learning scheme**

The optimization problem is formulated as the minimization of an energy term \( \mathcal{L} \) also called loss function which in our case will be defined as a simple mean-squared-error (\( \mathcal{L}_2 \))
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

$$\mathcal{L} [\mathcal{M}_{\text{NN}}(\theta_k), \nabla \cdot s] = \| \mathcal{M}_{\text{NN}}(\theta_k) - \nabla \cdot s \|_2$$  \hspace{1cm} (18)$$

where $\theta_k$ 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 sub-model $\mathcal{V}$ composed of six non-linear convolutions 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 sub-model $\mathcal{T}$ is built from a single convolution with a wider kernel of size $7 \times 7 \times 7$ cleared from any non-linearities. The minimization problem is carried using an Adam optimizer \cite{kingma2014adam} and an adaptable learning-rate linearly decreasing from $1e^{-4}$. As a pre-processing 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 \cite{goodfellow2016deep}.

The implementation using PyTorch is available at \cite{pytorch2019} 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 a zero variance over multiple runs of uniformly sampled $\lambda$ and $\beta$ coefficients. The soft constraint imposed for rotation invariance reduces the error variance from one order of magnitude over the 6 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.
TABLE II. Evaluation of the three additional constraints provided by $\mathcal{M}_{\text{SGTNN}}$; expectation and variance of the 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 $\lambda$ and $\beta$ are uniformly sampled values.

| $\mathcal{M}_{\text{NN}}(\mathbf{u}, \lambda \Phi)$ | $\mathcal{M}_{\text{NN}}(\mathbf{u} + \beta, \Phi)$ | $\mathcal{M}_{\text{NN}}(A_{ij} \mathbf{u}, \Phi)$ |
|---|---|---|
| $E[\mathcal{L}_2]$ | $\text{Var}[\mathcal{L}_2]$ | $E[\mathcal{L}_2]$ | $\text{Var}[\mathcal{L}_2]$ |
| 1.7251 | 0.0629 | 1.4643 | 0.0 |
| 1.8307 | 0.0373 | 1.4643 | 0.0 |
| 1.6022 | $1.9539 \cdot 10^{-6}$ | 1.4638 | $2.5107 \cdot 10^{-7}$ |

TABLE III. A priori evaluation of the SGS term in developed turbulence regime using testing data with the metrics described in Section V.1.

V.1. A priori evaluation

V.1.1. Developed turbulence regime

We first perform the evaluation on the developed regime of the testing dataset (see Fig. 2) which is the closest to the training data, and is expected to perform the best. In Table III we show the results of the a priori evaluation with the same range of values than 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 $L_2$ error and $\mathcal{P}$ compared to the $\mathcal{M}_{\text{CNN}}$. On the statistical metrics, it is clear from Fig. 5 that the $\mathcal{M}_{\text{MLP}}$ is better at reproducing mean values (around $\nabla \cdot \mathbf{s} = 0$) which tends to lead to a smaller $\mathcal{J}$ and $\mathcal{K}$. However, the $\mathcal{M}_{\text{SGTNN}}$ is more accurate on the tails
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.

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 \( M_{SGTNN} \) is in best agreement with the theoretical quantiles, for which a perfectly reproduced distribution would fit a straight line. 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 metrics \( \mathcal{L} \).

We see however that the \( M_{MLP} \) is not performing well when evaluated on the divergence, but 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 configurations will test the ability of the NN based models to generalize to flows that were not part of the training data.

V.1.2. 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 transitional regime. Fig. 6 shows that even if the \( L_2 \) is growing during the high-Reynolds number time-span, we can easily see that the NN models follow a similar behavior than in developed turbulence compared to the algebraic models. We note that the correlation given by \( P \) is quite low during the first iterations but quickly reaches its statistical mean at \( t \approx 5 \), as well as the metric \( J \) from which a large distance is observed especially from the \( M_{MLP} \).

We can also see in Fig. 7 that the first iterations of the \( M_{MLP} \) and \( M_{CNN} \) are producing a non-null SGS term while it is expected to be almost zero since the presence of turbulence
FIG. 6. Time evolution of the different metrics in transitional regime. Mean-squared-error (top left), Pearson’s coefficient (top right), Jensen-Shannon distance (bottom left) and integral dissipation error (bottom right).

is limited at that time. On the other side, the $M_{\text{SGTNN}}$ follows an expected behavior, since it gradually produces a SGS term that corresponds to the DNS. A difference that appears for each NN based models 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. [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 models.

V.1.3. Decaying scalar regime

In this last a priori configuration, we remove the scalar forcing while maintaining the velocity field in turbulent motion. This results in a decay, or mixing of the scalar within the domain. This regime is a difficult case for the models presented in this work and assess
FIG. 7. Data slices of the predicted SGS terms from NN models in transitional regime. It is particularly visible that $\mathcal{M}_{\text{MLP}}$ is producing a non-null SGS term during the first iterations, while $\mathcal{M}_{\text{CNN}}$ quickly produces large localized values.

their capacity to generalize to flow conditions that have not been seen during the training phase. Both $\mathcal{M}_{\text{MLP}}$ and $\mathcal{M}_{\text{CNN}}$ see their structural and statistical performance drastically decreasing after a short time $t \approx 5$, as seen in Fig. 8 on the $L_2$ and particularly on the correlation $\mathcal{P}$ and the distance $\mathcal{J}$. Interestingly, we observe that $\mathcal{M}_{\text{CNN}}$ is producing a dissipation error which does not tend toward zero compared to the other models.

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 configurations (first iterations in transitional regime and last iterations during scalar decay).
FIG. 8. Time evolution of the different metrics in decaying scalar regime. Mean-squared-error (top left), Pearson’s coefficient (top right), Jensen-Shannon distance (bottom left) and integral dissipation error (bottom right).

V.2. A posteriori evaluation

We now run new simulations using the ML 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 keep the velocity fields at DNS resolution, i.e. $512^3$, while the scalar fields of the different models evolve at the same resolution than the filtered training data, i.e. $64^3$. The prediction provided by the different models are able to close the filtered equation (4) such that $\mathcal{M}_{\text{NN}} = \nabla \cdot \mathbf{s}$ given the low resolution scalar and filtered velocities. Now we look at different statistical metrics such as the filtered scalar variance $\langle \Phi^2 \rangle - \langle \Phi \rangle^2$ and the resolved scalar enstrophy $\langle \mathbf{\omega}^2 \rangle / 2 = \langle \nabla^2 \Phi \rangle$ which is particularly interesting to characterizes the energy transfers in the smallest resolved scales [47], from which the SGS model is the most important.
FIG. 9. Data slices of the predicted SGS terms from NN models in scalar decay regime. It is particularly visible that $M_{\text{MLP}}$ and $M_{\text{CNN}}$ are still predicting a non-null SGS term when the scalar is completely mixed.

V.2.1. Developed turbulence regime

In the first two simulations, we start in a developed turbulence regime at $S = 20000$ of the training data (see Fig. 1) and the testing data (see Fig. 2) with the same velocity and scalar forcing than used in the filtered dataset. Using the same starting point than the training data is useful in order to consider the accumulation of errors during the simulation, since the models only learnt from a finite number of samples of the exact simulation. We see in Fig. 10 that the scalar variance, on the left, produced by the $M_{\text{SGTNN}}$ and the $M_{\text{CNN}}$ are the closest from the DNS except after a long integration time, $t \approx 20$ where the $M_{\text{CNN}}$ starts accumulating error. The behavior of the model on the largest scales is shown by the
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).

resolved scalar enstrophy on the right. Here, the $M_{MLP}$ has a consistent behavior almost similar to the $M_{SGTNN}$. 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 [25] is most effective close to the spectral filter at $k \geq 30$. Now, the same conclusions can be drawn with the testing data starting point as depicted in Fig. 12 and 13. We can already see that a CNN without physical knowledge is not able to reproduce the dynamics of the SGS term in similar conditions than those of the training data. Indeed, it performs at the same accuracy for the scalar variance and worse for the resolved scalar enstrophy than some algebraic models. The effect of the physical invariances is primordial and give the ability to get the best performances from the neural network.
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).

V.2.2. Transitional regime

The transitional regime is already testing the extrapolation capabilities of the model. The difficulty of this configuration resides in the transition to turbulence, which then comes down to the previous configuration 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 same configuration seems to correlate with Fig. 15, 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. 16 indicates that the spectrum 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 deviation appear during the transition to turbulence, i.e. the first iterations.
FIG. 14. Statistics of simulation in transitional regime starting from a newly initialized scalar field. Scalar variance (left), resolved scalar enstrophy (right).

FIG. 15. Normalized errors $N_{err}(x, \hat{x}) = (x - \hat{x})/\hat{x}$ on the statistics of simulation in transitional regime starting from a newly initialized scalar field. Scalar variance (left), resolved scalar enstrophy (right).

of the simulation.

V.2.3. Decaying scalar regime

We saw in a priori tests that the $\mathcal{M}_{SGTNN}$ gave the best performances in the decaying scalar configuration while the other NN models were not able to generalize. In a posteriori, we observe the same behavior in Fig. 17 and 18. More precisely, the resolved scalar enstrophy shows that the $\mathcal{M}_{MLP}$ and $\mathcal{M}_{CNN}$ are still producing energy transfers after a long time. Moreover, we see in the spectrum (Fig. 19) 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
FIG. 16. Energy spectrum of simulation in transitional regime starting from a newly initialized scalar field at $t = 6$ (left) and $t = 25$ (right).

FIG. 17. Statistics of simulation in decaying scalar regime starting from testing data. Scalar variance (left), resolved scalar enstrophy (right).

simulation after $t = 16$. The $\mathcal{M}_{\text{SGTNN}}$ however is still in excellent agreement with the DNS and out-performs significantly the algebraic models, even in configurations that were not part of the training data.

VI. SUMMARY

We propose a new 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 out-performs classical algebraic models on different metrics. During the a posteriori evaluation, we demonstrate the capabilities of the model to generalize to unseen flow configurations compared to NN models that do not embed physical knowledge.
FIG. 18. Normalized errors $N_{\text{err}}(x, \hat{x}) = (x - \hat{x})/\hat{x}$ on the statistics of simulation in decaying scalar regime starting from testing data. Scalar variance (left), resolved scalar enstrophy (right).

FIG. 19. Energy spectrum of simulation in decaying scalar regime starting from testing data at $t = 8$ (left) and $t = 16$ (right).

Thus, we think that to be applicable to large scale simulations, NN models will require some form of hard physical knowledge embedding. However, 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.

VII. ACKNOWLEDGEMENTS

This work is supported by the French Centre National de la Recherche Scientifique (CNRS) through the 80—PRIME project. Computations were performed using high-
performance facilities HPC resources from GENCI (Grant No. 020611) and GRICAD infrastructures.

[1] K. Duraisamy, G. Iaccarino, and H. Xiao, Annual Review of Fluid Mechanics 51, 357 (2019).
[2] R. Wang, K. Kashinath, M. Mustafa, A. Albert, and R. Yu, arXiv preprint arXiv:1911.08655 (2019).
[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, et al., arXiv preprint arXiv:1911.05180 (2019).
[4] A. T. Mohan, N. Lubbers, D. Livescu, and M. Chertkov, in ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations (2020).
[5] M. Gamahara and Y. Hattori, Physical Review Fluids 2, 054604 (2017).
[6] C. Xie, J. Wang, and E. Weinan, Physical Review Fluids 5, 054606 (2020).
[7] R. Maulik, O. San, A. Rasheed, and P. Vedula, Journal of Fluid Mechanics 858, 122 (2019).
[8] C. Xie, J. Wang, K. Li, and C. Ma, Physical Review E 99, 053113 (2019).
[9] A. Beck, D. Flad, and C.-D. Munz, Journal of Computational Physics 398, 108910 (2019).
[10] S. Pawar, O. San, A. Rasheed, and P. Vedula, Theoretical and Computational Fluid Dynamics , 1 (2020).
[11] C. J. Lapeyre, A. Misdariis, N. Cazard, D. Veynante, and T. Poinsot, Combustion and Flame 203, 255 (2019).
[12] T. Bolton and L. Zanna, Journal of Advances in Modeling Earth Systems 11, 376 (2019).
[13] J. Ling, A. Kurzawski, and J. Templeton, Journal of Fluid Mechanics 807, 155 (2016).
[14] J. Ling, R. Jones, and J. Templeton, Journal of Computational Physics 318, 22 (2016).
[15] M. Bode, M. Gauding, Z. Lian, D. Denker, M. Davidovic, K. Kleinheinz, J. Jitsev, and H. Pitsch, arXiv preprint arXiv:1911.11380 (2019).
[16] P. Sagaut, Large eddy simulation for incompressible flows: an introduction (Springer, 2006).
[17] J. Smagorinsky, Monthly Weather Review 91, 99 (1963).
[18] R. A. Clark, J. H. Ferziger, and W. C. Reynolds, Journal of Fluid Mechanics 91, 1 (1979).
[19] J. Bardina, J. Ferziger, and W. Reynolds, in 13th fluid and plasmadynamics conference (1980) p. 1357.
[20] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot, Physics of Fluids A: Fluid Dynamics 3, 1760 (1991).
[21] D. K. Lilly, Physics of Fluids A: Fluid Dynamics 4, 633 (1992).
[22] G. Balarac, J. Le Sommer, X. Meunier, and A. Volland, Physics of Fluids 25, 075107 (2013).
[23] Y. LeCun, Y. Bengio, and G. Hinton, Nature 521, 436 (2015).
[24] A. Volland, G. Balarac, and C. Corre, Journal of Turbulence 18, 854 (2017).
[25] G. Portwood, B. Nadiga, J. Saenz, and D. Livescu, arXiv preprint arXiv:2004.07207 (2020).
[26] G. K. Batchelor, Journal of Fluid Mechanics 5, 113 (1959).
[27] A. M. Obukhov, Izvestiia Akademii Nauk SSSR, Ser. Geogr. i Geofiz. 13, 58 (1970).
[28] S. Corrsin, Journal of Applied Physics 22, 469 (1951).
[29] K. Alvelius, Physics of Fluids 11, 1880 (1999).
[30] P. Moin, K. Squires, W. Cabot, and S. Lee, Physics of Fluids A: Fluid Dynamics 3, 2746 (1991).
[31] A. Leonard et al., Advances in Geophysics 18, 237 (1974).
[32] C. Meneveau and J. Katz, Annual Review of Fluid Mechanics 32, 1 (2000).
[33] D. M. Endres and J. E. Schindelin, IEEE Transactions on Information Theory 49, 1858 (2003).
[34] J. Hodges, Arkiv för Matematik 3, 469 (1958).
[35] L. Berselli, T. Iliescu, and W. Layton, Mathematics of large eddy simulation of turbulent flows (Springer, 2006).
[36] In the deep learning literature, fully-connected architectures refer to NN architectures based on dense or fully-connected layers. The MLP is an example of fully-connected architectures.
[37] C. G. Speziale, Journal of Fluid Mechanics 156, 55 (1985).
[38] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, IEEE Transactions on Industrial Electronics 63, 7067 (2016).
[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, in Advances in neural information processing systems (2012) pp. 1097–1105.
[40] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker, Medical Image Analysis 36, 61 (2017).
[41] M. Oberlack, Annual Research Briefs , 3 (1997).
[42] T. Cohen and M. Welling, in International conference on machine learning (2016) pp. 2990–2999.
[43] M. Weiler, M. Geiger, M. Welling, W. Boomsma, and T. S. Cohen, in Advances in Neural Information Processing Systems (2018) pp. 10381–10392.

[44] D. P. Kingma and J. Ba, arXiv preprint arXiv:1412.6980 (2014).

[45] S. Ioffe and C. Szegedy, arXiv preprint arXiv:1502.03167 (2015).

[46] H. Frezat, G. Balarac, J. Le Sommer, R. Fablet, and R. Lguensat, Subgrid transport neural network (2020).

[47] O. San, A. E. Staples, and T. Iliescu, International Journal of Computational Fluid Dynamics 29, 40 (2015).