MIT Open Access Articles

Machine-learning energy-preserving nonlocal closures for turbulent fluid flows and inertial tracers

Machine-learning energy-preserving nonlocal closures for turbulent fluid flows and inertial tracers

Alexis-Tzianni G. Charalampopoulos, Themistoklis P. Sapsis*

Department of Mechanical Engineering, Massachusetts Institute of Technology

February 16, 2021

Abstract

We formulate a data-driven, physics-constrained closure method for coarse-scale numerical simulations of turbulent fluid flows. Our approach involves a closure scheme that is non-local both in space and time, i.e. the closure terms are parametrized in terms of the spatial neighborhood of the resolved quantities but also their history. The data-driven scheme is complemented with a physical constrain expressing the energy conservation property of the nonlinear advection terms. We show that the adoption of this physical constrain not only increases the accuracy of the closure scheme but also improves the stability properties of the formulated coarse-scale model. We demonstrate the presented scheme in fluid flows consisting of an incompressible two-dimensional turbulent jet. Specifically, we first develop one-dimensional coarse-scale models describing the spatial profile of the jet. We then proceed to the computation of turbulent closures appropriate for two-dimensional coarse-scale models. Training data are obtained through high-fidelity direct numerical simulations (DNS). We also showcase how the developed scheme captures the coarse-scale features of the concentration fields associated with inertial tracers, such as bubbles and particles, carried by the flow but not following the flow. We thoroughly examine the generalizability properties of the trained closure models for different Reynolds numbers, as well as, radically different jet profiles from the ones used in the training phase. We also examine the robustness of the derived closures with respect to the grid size. Overall the adoption of the constraint results in an average improvement of 26% for one-dimensional closures and 29% for two-dimensional closures, being notably larger for flows that were not used for training.

1 Introduction

Turbulent fluid flows in nature and engineering are characterized by a wide range of spatial and temporal scales with nonlinear interactions making their reduced order modeling a challenging task. Over the last decades several ideas have emerged that successfully model turbulent fluid flows, such as Large Eddy Simulations [23, 29]. However, these methods still require very high resolution in order to satisfactory model the large scale dynamics, as well as features associated with those. This is an important computational obstacle especially for applications involving uncertainty quantification, optimization, and risk analysis where there is a need for a large number of accurate simulations.

The recent machine-learning advances has sparked a new interest in utilizing deep neural networks to develop reduced order models for turbulent flows. The machine-learning closures abandon the

---

*Corresponding author: sapsis@mit.edu, Tel (617) 324-7508, Fax: (617) 253-8689

path of a closed-form expression for the closure terms into utilizing experimental or costly high-fidelity computations to train a neural network and predict the nonlinear energy transfers between resolved and unresolved scales. One of the first such effort of deep neural networks to turbulent flows appeared in [22], where nonlinear autoencoders were utilized to reconstruct the near wall field in a turbulent flow. Since then there has been a plethora of efforts focusing on machine-learning closures using different data-driven schemes, such as artificial neural networks in fluid flows [10, 35, 20, 46] and multiphase flows [17, 18], random forest regressions [45], spatially nonlocal schemes such as convolutional neural networks [47], stochastic data-driven representations using generative adversarial networks [36], reinforcement learning [24] and with applications ranging from engineering turbulence to geophysics and beyond (see [6] for a recent review).

One of the great advantage of machine-learning closures is their capability to seamlessly model non-locality in time. Indeed, there is no a priori reason to expect that the closure terms of a complex system will behave in a Markovian manner, i.e. depend only on the current reduced-order state of the system. In the contrary, Takens embedding theorem [37] states that if we observe only a limited number of the state variables of a system, in principle, we can still obtain the attractor of the full system by using delay embedding of the observed state variables. Therefore, it is essential to incorporate memory effects when we model closure terms for turbulent fluid flows. This approach has found success in a number of physical applications involving bubble motion and multiphase flows [42, 7], as well as the reduced-order modeling of chaotic dynamical systems [39, 44, 41].

On the other hand, machine-learning schemes allows us to parametrize the closure terms using a large number of input variables opening the possibility for non-local models in space (see [47] for an application to the advection of a passive scalar). Spatially, non-local models have been advocated for turbulent closures and there is a plethora of related ideas ranging from scale-dependent closures [25], non-local Reynolds Stress models [13], and fractional-operators closures [30]. Several ideas related to functional neural networks or operator neural networks have shown great promise on this direction [8, 9], but not yet utilized in the context of turbulent closure models.

Beyond local closures deep neural networks have also been used successfully in combination with the underlying governing equations for reconstructing complex fluid flows and identifying flow parameters. Specifically, physics-informed neural networks [26] identify the optimal solution (either the flow itself or parameters associated with it) by minimizing an objective function that contains the Navier-Stokes equations, as well as scattered data in space and time. Inclusion of the governing equations significantly improves the behavior of the data-driven scheme, while the representation of the solution in terms of a neural networks circumvents the need for a grid or spatial discritization scheme. The method has shown great promise for reconstructing fluid flows given spatio-termporal measurements [28], as well as recovering macroscopic quantities such as lift or drag for vortex-induced vibration problems [27]. Previous efforts along this line include the embedding of symmetries such as Galilean invariance to the neural net predictions for an anisotropic Reynolds stress tensor [16, 15].

Our aim in this work is to formulate energy-preserving spatio-temporally non-local turbulent closures which are a priori consistent with the conservation properties of the advection term in Navier-Stokes equations. Specifically, we utilize machine learning schemes which represent the effect of the small scales at each spatial location, using as input the large scale features of the flow in a spatial neighborhood of this location. Past values of the large scale features are also employed as inputs for the turbulent closures in a causal manner. These data-driven schemes are enforced to be consistent with physical constraints expressing the energy exchanges between resolved and unresolved scales. These constraints follow from the energy-conserving properties of the nonlinear advection operator in Navier-Stokes and have been utilized previously in the context of uncertainty quantification and stochastic closure models [31, 32, 33, 19]. In contrast to previous effort where the full system equation is used as a constraint, assuming perfect knowledge of the equation form and/or

parameters (e.g. [28]), the formulated constraint in this work expresses a universal property of the advection terms, i.e. that they do no create or destroy kinetic energy of the flow. To improve the stability properties of the computed closures we also employ imitation learning [1].

We first formulate the objective function used in the training phase. This step also includes the physical constraint and its derivation using Gauss theorem. We subsequently consider a forced two-dimensional jet flow. We first take into account the invariance of the flow in one direction to derive one-dimensional machine-learned closures using DNS information. As a next step, we apply the method on the computation of two-dimensional turbulent closures that do not rely on this special symmetry. We compare the obtained coarse-scale model with DNS and assess its generalizability properties for different Reynolds numbers, as well as different jet profiles which have not been used in the training phase. We thoroughly examine the role of the physical constraint on the stability properties and accuracy of the coarse-scale equations. In addition, we assess our closure scheme on capturing the evolution of concentration for inertial tracers, such as bubbles and aerosols.

2 Formulation of energy-preserving closure schemes

Our aim is to derive Eulerian, data-driven closure schemes for turbulent fluid flows, as well as for inertial tracers advected by those. These closure schemes will not only rely on DNS training data, but also on the physical constraint that follows from the energy conservation principles that the nonlinear advection terms satisfy [31, 19]. The effectiveness of the closure schemes is assessed by how well the coarse-scale equations can reproduce the mean flow characteristics for problems that reach a statistical equilibrium. Higher order closures may be utilized to improve predictions for higher order statistics such as the flow spectrum. However, in this work we will focus on closures that aim to model the mean flow characteristics.

First, we introduce a spatial-averaging operator that will define the coarse scale version of the quantities of interest and their evolution equations. Specifically, we decompose any field of interest f as

$$f = \overline{f} + f', \tag{1}$$

where $\overline{f}$ corresponds to the large-scale component of the quantity and f' corresponds to the small-scale component. As a result we always have $\overline{f'} = 0$.

2.1 Averaged Navier-Stokes equations

We consider the Navier-Stokes equations in dimensionless form:

$$\frac{D\mathbf{u}}{Dt} = -\nabla p + \frac{1}{\mathrm{Re}}\Delta\mathbf{u} + \nu\nabla^{-4}\mathbf{u} + \mathbf{F}, \tag{2}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{3}$$

where $\mathbf{u}$ is the velocity field of the fluid, p its pressure, Re is the Reynolds number of the flow, $\frac{D}{Dt}$ is the material derivative operator and $\mathbf{F}$ denotes an external forcing term. Parameter ν is a hypoviscosity coefficient aiming to remove energy from large scales and maintain the flow in a turbulent regime. Using the decomposition (1) into the fluid flow eqs. (2) and applying the averaging operator we obtain:

$$\partial_t\overline{\mathbf{u}} = -\overline{\mathbf{u}}\cdot\nabla\overline{\mathbf{u}} - \overline{\mathbf{u}'\cdot\nabla\mathbf{u}'} - \nabla\overline{p} + \frac{1}{\mathrm{Re}}\Delta\overline{\mathbf{u}} + \nu\nabla^{-4}\overline{\mathbf{u}} + \overline{\mathbf{F}}, \tag{4}$$

3

$$\nabla \cdot \overline{\mathbf{u}} = 0. \tag{5}$$

Clearly, the averaged evolution equations do not comprise a closed system anymore due to the nonlinearity of the advection term. As a result, the term $\overline{\mathbf{u}' \cdot \nabla \mathbf{u}'}$ is not defined by the evolution equations and needs to be parametrized.

2.2 Averaged advection equation for inertial tracers

One can follow a similar process for the advection equation governing the motion of small inertial tracers. In particular for small inertia particles their Lagrangian velocity, $\mathbf{v}$, is a small perturbation of the underlying fluid velocity field [12]:

$$\mathbf{v} = \mathbf{u} + \epsilon \left(\frac{3R}{2} - 1 \right) \frac{D\mathbf{u}}{Dt} + O(\epsilon^2), \quad \epsilon = \frac{St}{R} \ll 1, \quad R = \frac{2\rho_f}{\rho_f + 2\rho_p}, \tag{6}$$

where $\epsilon = \frac{St}{R} \ll 1$ is a parameter representing the importance of inertial effects, St is the particle or bubble Stokes number, and $R = \frac{2\varrho_f}{\varrho_f + 2\varrho_p}$ is a density ratio with ϱ_p and ϱ_f being the density of the particle or bubble and the flow respectively.

By introducing ρ as the concentration of tracers at a particular point, we can write the following transport equation

$$\partial_t \rho + \nabla \cdot (\mathbf{v}\rho) = \nu_2 \Delta^4 \rho. \tag{7}$$

The right-hand-side of the transport equation represents a hyperviscosity term. Introducing the decomposition of eq. (1) in the evolution eqs. (6) and (7), we obtain

$$\overline{\mathbf{v}} = \overline{\mathbf{u}} + \epsilon \left(\frac{3R}{2} - 1 \right) \left(\partial_t \overline{\mathbf{u}} + \overline{\mathbf{u}} \cdot \nabla \overline{\mathbf{u}} + \overline{\mathbf{u}' \cdot \nabla \mathbf{u}'} \right), \tag{8}$$

$$\partial_t \overline{\rho} + \nabla \cdot (\overline{\mathbf{v}}\overline{\rho}) + \overline{\nabla \cdot (\mathbf{v}'\rho')} = \nu_2 \Delta^4 \overline{\rho}. \tag{9}$$

Once again, the closure term $\overline{\nabla \cdot (\mathbf{v}'\rho')}$ appears, which requires parametrization.

2.3 Data-driven parametrization of the closure terms

While the full Navier-Stokes equations and the associated advection equations are Markovian and spatially-local, i.e. the evolution of the flow or concentration field in a specific location and time instant depends only on the current time instant and the current neighborhood, this is not necessarily the case for the averaged version of these equations. In particular, for the averaged equations we typically do not have access to the full-state information, required to fully describe the evolution of the system. In this case the missing information is the small-scale dynamics.

To overcome this limitation we recall Takens embedding theorem [37], which states that if we observe only a limited number of the state variables of a system, in principle, we can still obtain the attractor of the full system by using delay embeddings of the observed state variables. In other words, under appropriate conditions there is a map between the delays of the observed state variables and the full state system. Although the theorem itself is several decades old, we can now rely on recently developed data-driven schemes that can implement such mapping as part of their training process, enhancing the accuracy of predictions (see e.g. [40, 43]). To this end, we parametrize the closure

4

terms with *non-local in time* (but still causal) representations, based on Long-Short-Term Memory (LSTM) recurrent neural networks (RNN) and Temporal Convolutional Networks (TCN) [14]. The specific RNN implementation was picked based on its tested ability to incorporate long-term memory effects of hundreds of time-delays, while simpler RNN models suffer from vanishing or exploding gradients [2].

Beyond non-locality in time we also choose to employ non-locality in space. That is, given a point in space $\mathbf{x}$, we use information from points that lie in a small neighborhood of $\mathbf{x}$. Clearly, incorporating information from the entirety of the domain is not only computationally infeasible but also redundant and can lead to stability issues. For this reason we use convolutions in space to make sure that we incorporate information only from a region around each point and not from the entire domain. The parameterization is based on a stacked LSTM architecture [11], which utilizes LSTM layers with the detail that all input and recurrent transformations are convolutional.

As a result, the closure terms are modeled in the following form:

$$
\begin{aligned}
\overline{\mathbf{u}' \cdot \nabla \mathbf{u}'}(\mathbf{x}, t) &= \mathbb{D}_{\mathbf{u}} \left[\theta_1; \overline{\xi}[\alpha(\mathbf{x}), \chi(t)] \right], \\
\overline{\nabla \cdot (\mathbf{v}' \rho')}(\mathbf{x}, t) &= \mathbb{D}_{\rho} \left[\theta_2; \overline{\zeta}[\alpha(\mathbf{x}), \chi(t)] \right],
\end{aligned}
\tag{10}
$$

where $\overline{\xi}$ and $\overline{\zeta}$ are (averaged) flow features to be selected, $\alpha(\mathbf{x})$ denotes a pre-selected neighborhood of points around $\mathbf{x}$ over which the averaged state is considered, i.e. $\alpha(\mathbf{x}) = \{\mathbf{x}, \mathbf{x}+\mathbf{x}_1, \mathbf{x}+\mathbf{x}_2, ..., \mathbf{x}+\mathbf{x}_N\}$, and $\chi(t)$ denotes the history of the averaged state backwards from time t, i.e. $\chi(t) = \{t, t-\tau_1, ..., t-\tau_2, ..., t-\tau_N\}$. The vectors θ_1 and θ_2 denote the hyperparameters of the neural networks and their optimization is performed empirically. The spatial neighborhood, $\alpha(\mathbf{x})$, is selected such that if we further increase it, the training error does not significantly reduce any more. Note however, that the number of points in space that have to be considered in $\alpha(\mathbf{x})$ is dependent on the discretization of the domain, i.e., if we increase the resolution of our model the number of neighborhood points in $\alpha(\mathbf{x})$ should increase respectively. In the application section we study the effect of spatial discretization. We use a similar approach for the temporal history, $\chi(t)$.

2.3.1 Temporal integration

We point out that our numerical goal is inline prediction. This means that the neural nets described by eq. (10) must be coupled with the evolution eqs. (4)-(9). For a simple forward Euler scheme for temporal integration, this would imply that by knowing the values of $\overline{\mathbf{u}}, \overline{\rho}$ at time t we can predict the closure terms at time t using eq. (10) and use their values to integrate eqs. (4)-(9) by one time-step δt so that we compute $\overline{u}_1, \overline{\rho}$ at time $t + \delta t$. However, if we want to use a higher-order integration scheme like a 4th-order explicit Runge-Kutta, we need to evaluate the closure terms at time $t + \delta t/2$ as well. Since, we do not have access to the required time-history for such a prediction, we instead integrate in time not by δt but by $2\delta t$ and thus get a time-integration error of the for $O[(2\delta t)^4]$.

2.4 Physical constraints

An important feature of our data-driven closure schemes is the requirement to satisfy certain physical principles. Specifically, we utilize the energy flux constraint that the advection term does not alter the total kinetic energy of the model [31, 32]. This constraint, follows from Gauss identity. Specifically, for any scalar fielsd, α, β, and divergence-free field, Φ, we have from Gauss identity:

$$
\int_{\Omega} \frac{\partial \alpha}{\partial x_j} \beta \Phi_j d\mathbf{x} = - \int_{\Omega} \frac{\partial \beta}{\partial x_j} \alpha \Phi_j d\mathbf{x} + \int_{\partial \Omega} \alpha \beta \Phi_j n_j d\mathbf{x},
\tag{11}
$$

5

where n_j is the unit vector on the boundary, $\partial\Omega$. Applying the above for $\alpha = \beta = u_k$ and $\Phi_j = u_j$, we obtain the general three-dimensional constraint:

$$\int_\Omega \mathbf{u} \cdot (\mathbf{u} \cdot \nabla)\mathbf{u}\,\mathrm{d}\mathbf{x} = \int_{\partial\Omega} \mathcal{E}\,\mathbf{u} \cdot \mathbf{n}\,d\mathbf{x}, \quad \mathcal{E} = \frac{1}{2}\mathbf{u} \cdot \mathbf{u} \tag{12}$$

where Ω is the domain in which the fluid flow is defined. The above constraint essentially expresses the fact that the nonlinear advection terms do not change the total kinetic energy of the system. In what follows we will consider the case of periodic boundary conditions, where the right hand side in (12) vanishes. However, the same ideas are applicable for arbitrary boundary conditions. We apply the decomposition (1) and the spatial averaging operator to this equation and obtain:

$$\int_\Omega \overline{\mathbf{u}} \cdot (\overline{\mathbf{u}} \cdot \nabla)\overline{\mathbf{u}}\,\mathrm{d}\mathbf{x} + \int_\Omega \overline{\mathbf{u}} \cdot \overline{(\mathbf{u}' \cdot \nabla)\mathbf{u}'}\,\mathrm{d}\mathbf{x} + \int_\Omega \overline{\mathbf{u}' \cdot (\overline{\mathbf{u}} \cdot \nabla)\mathbf{u}'}\,\mathrm{d}\mathbf{x}$$
$$+ \int_\Omega \overline{\mathbf{u}' \cdot (\mathbf{u}' \cdot \nabla)\overline{\mathbf{u}}}\,\mathrm{d}\mathbf{x} + \int_\Omega \overline{\mathbf{u}' \cdot (\mathbf{u}' \cdot \nabla)\mathbf{u}'}\,\mathrm{d}\mathbf{x} = 0, \tag{13}$$

From the last equation we have the *physical constraint* that the closure term $\mathbb{D}_{\mathbf{u}}$ must satisfy

$$\int_\Omega \overline{\mathbf{u}} \cdot \mathbb{D}_{\mathbf{u}} \left[\theta_1; \overline{\xi}[\alpha(\mathbf{x}), \chi(t)]\right] d\mathbf{x} = A[\mathbf{u}] \triangleq -\int_\Omega \overline{\mathbf{u}} \cdot (\overline{\mathbf{u}} \cdot \nabla)\overline{\mathbf{u}}\,\mathrm{d}\mathbf{x} - \int_\Omega \overline{\mathbf{u}' \cdot (\overline{\mathbf{u}} \cdot \nabla)\mathbf{u}'}\,\mathrm{d}\mathbf{x}$$
$$- \int_\Omega \overline{\mathbf{u}' \cdot (\mathbf{u}' \cdot \nabla)\overline{\mathbf{u}}}\,\mathrm{d}\mathbf{x} - \int_\Omega \overline{\mathbf{u}' \cdot (\mathbf{u}' \cdot \nabla)\mathbf{u}'}\,\mathrm{d}\mathbf{x}, \tag{14}$$

where $A[\mathbf{u}]$ is a function that depends on the training data and the discretization. Such a constraint can be added to the training process in a straightforward manner through the objective function. We emphasize that one could formulate a physical constraint based e.g. on the Navier-Stokes equations directly. However, this assumes exact knowledge of the flow-specifics. This is not the case here since the above constraint expresses a universal property, i.e. that advection terms do not create or destroy energy.

2.5 Objective function for training

In terms of the training process itself, we normalize the input and output data as usually suggested (see e.g. [34]). The loss function for this problem is chosen to be the single-step prediction mean square error superimposed with the physical constraint. This can be formulated as

$$L(\theta_1) = \int_{\Omega \times T} \left|\left| \mathbb{D}_{\mathbf{u}} \left[\theta_1; \overline{\xi}\right] - \overline{(\mathbf{u}' \cdot \nabla)\mathbf{u}'} \right|\right|^2 d\mathbf{x}dt + \lambda \int_T \left| \int_\Omega \overline{\mathbf{u}} \cdot \mathbb{D}_{\mathbf{u}} \left[\theta_1; \overline{\xi}\right] d\mathbf{x} - A[\mathbf{u}] \right| dt \tag{15}$$

On the other hand, for the advection equation we have the objective function:

$$L(\theta_2) = \int_{\Omega \times T} \left|\left| \mathbb{D}_\rho \left[\theta_2; \overline{\zeta}[\alpha(\mathbf{x}), \chi(t)]\right] - \nabla \cdot \left(\overline{\mathbf{v}'\rho'}\right) \right|\right|^2 d\mathbf{x}dt. \tag{16}$$

An important question is which flow features are important as input for each of the two models. We examine this issue in detail in the following sections.

2.6 Imitation learning

While we use the single-step prediction for training, our aim is to use these models for multi-step prediction. Any such predictor introduces errors and these compounding errors change the input distribution for future prediction steps, breaking the train-test i.i.d assumption that is common in supervised learning. Under these circumstances, the error can be shown to grow exponentially [38]. This effect was observed for our setup as well, with the averaged equations becoming unstable. To alleviate this problem we use a version of imitation learning presented in [38], the Data as Demonstrator (DAD) method. Specifically, after we undergo a round of initial training we make predictions and demand that after a certain number of time-steps the prediction should again match the training data. Assuming an error tolerance δ_1, δ_2 for $\mathbb{D}_{\mathbf{u}}$ and $\mathbb{D}_\rho$ respectively, we can use Algorithm 1 as shown below. Note that in this pseudo-algorithm, quantities denoted with $*$ correspond to predictions of the neural network under the assumptions introduced by the DAD algorithm. By repeating this process, we create additional training data which we use together with the original training data to continue training the model. We follow the exact same approach for the closure term $\overline{\nabla \cdot (\mathbf{v}'\rho')}$ as well. We repeated this process 20 times, in order to achieve sufficient robustness for our models (see Algorithm 1).

Algorithm 1: Training of closure scheme

> **Input:** Reference closure terms $\{ \mathbb{D}_i^{\mathrm{DNS}}(t_0), ..., \mathbb{D}_i^{\mathrm{DNS}}(t_p)\}$ $i = \mathbf{u}, \rho$, computed from DNS
> **Data:** NN architecture, averaging-operator, discretization, δt and input flow features $\overline{\mu}$
> **Result:** Predicted closure terms $\{ \mathbb{D}_i^{\mathrm{ML}}(t_0), ..., \mathbb{D}_i^{\mathrm{ML}}(t_p)\}$ $i = \mathbf{u}, \rho$.

1 Set $\mu^{\mathrm{Ref}} = \mu^{\mathrm{DNS}}$ and $\mathbb{D}_{\mathbf{u}}^{\mathrm{Ref}} = \mathbb{D}_{\mathbf{u}}^{\mathrm{DNS}}$;
2 Train $\mathbb{D}_{\mathbf{u}}^{\mathrm{ML}}$ using $\mathbb{D}_{\mathbf{u}}^{\mathrm{Ref}}$;
3 for $i = 1, ..., 20$ **do**
4   **for** $t = 0, \delta t, ..., T\delta t$ **do**
5 $s \rightarrow t$;
6   **while** $\left\| \mathbb{D}_{\boldsymbol{u}}^{ML}(s) - \mathbb{D}_{\boldsymbol{u}}^{DNS}(s) \right\| > \delta_1$ **do**
7   Predict $\overline{\mathbf{u}}^{\mathrm{ML}}(s + \delta t)$;
8   Predict $\mathbb{D}_{\mathbf{u}}^{\mathrm{ML}}(s + \delta t)$;
9 $s \rightarrow s + \delta t$;
10   **end**
11   Compute $\mathbb{D}_{\mathbf{u}}^*(s)$ so that $\overline{\mathbf{u}}^{\mathrm{ML}}(s + \delta t) = \overline{\mathbf{u}}^{\mathrm{DNS}}(s + \delta t)$;
12   Set $(\mu^{\mathrm{Ref}}, \mathbb{D}_{\mathbf{u}}^{\mathrm{Ref}}) = (\mu^{\mathrm{DNS}}, \mathbb{D}_{\mathbf{u}}^{\mathrm{ML}}) \cup (\mu^*|_s, \mathbb{D}_{\mathbf{u}}^*(s))$;
13   **end**
14   Train $\mathbb{D}_{\mathbf{u}}^{\mathrm{ML}}$ using $\mathbb{D}_{\mathbf{u}}^{\mathrm{Ref}}$;
15 end

---

3 Fluid flow setup

For the validation and assessment of the formulated closures we consider a two-dimensional turbulent jet where bubbles are also advected as passive inertial tracers. Specifically, the velocity field governing the bubbles is different from that of the underlying fluid flow (due to inertia effects), but the bubbles do not affect in any way the underlying fluid flow.

We setup a turbulent jet that fluctuates around a steady-state jet solution, $\mathbf{u}_{jet}$. In its dimen-

sionless form this system of equations can be written as

$$\frac{D\mathbf{u}}{Dt} = -\nabla p + \frac{1}{\text{Re}}\Delta\mathbf{u} + \nu\nabla^{-4}(\mathbf{u} - \mathbf{u}_{jet}) + \mathbf{F}, \tag{17}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{18}$$

where $\mathbf{u} = (u_1, u_2)$ and $Re = O(10^3)$. The domain is assumed rectangular, doubly periodic, i.e. $\mathbf{x} = (x, y) \in S^2 = [0, 2\pi] \times [0, 2\pi]$. For initial conditions, since we desire anisotropy in our flow, we use Gaussian jet structures of the general form

$$u_{1,jet} = \sum_i A_i \exp\Big[-c_i(y - \beta_i)^2\Big], \quad u_{2,jet} = 0, \tag{19}$$

where A_i, c_i, β_i are parameters. The role of the external forcing term, $\mathbf{F}$, is twofold: i) it contains a large-scale component to maintain the jet structure, by balancing the dissipation term, and ii) it has a small-scale and small-amplitude component to perturb the flow and trigger instabilities so that the we enter a turbulent regime. To achieve turbulence we choose a forcing term that acts only on a specific waveband with $6 \leq \|\mathbf{k}\| \leq 7$. Exciting a flow with a forcing localized only in a narrow wavenumber interval is common practice in the turbulence literature [5, 3, 21, 4].

Therefore, we adopt a form $\mathbf{F} = -\frac{1}{R}\Delta\boldsymbol{u}_{jet} + f$, with f being

$$f(\mathbf{x}, t) = \sum_i A_i(t) \cos(\mathbf{k} \cdot \mathbf{x} + \omega_i), \tag{20}$$

where $6 \leq |\mathbf{k}| \leq 7$, $A_i(t)$ are random vectors that follow a Gaussian white noise distribution (each one independent from the other) and ω_i are phases sampled from a uniform distribution over $[0, 2\pi]$. The standard deviation for these amplitudes is set to 0.03. This ensures that the energy and enstrophy inputs are localized in Fourier space and only a limited range of scales around the forcing is affected by the details of the forcing statistics. Furthermore, such a forcing ensures that the system reaches a jet-like statistical steady state after a transient phase. Due to the small-scale forcing being essentially homogeneous in space we can deduce that the statistical steady state profile is only dependent on y (since our large-scale forcing and initial conditions depend only on y). We solve this flow using a spectral method and 256^2 modes.

For the bubbles we use the perturbed advection field (eq. (6)) and the corresponding advection equation (7). For the simulations presented we use the inertial parameters, $\epsilon = 0.05$ and $R = 2$, which correspond to small bubbles. A typical snapshot of the described flow can be seen in Figure 1.
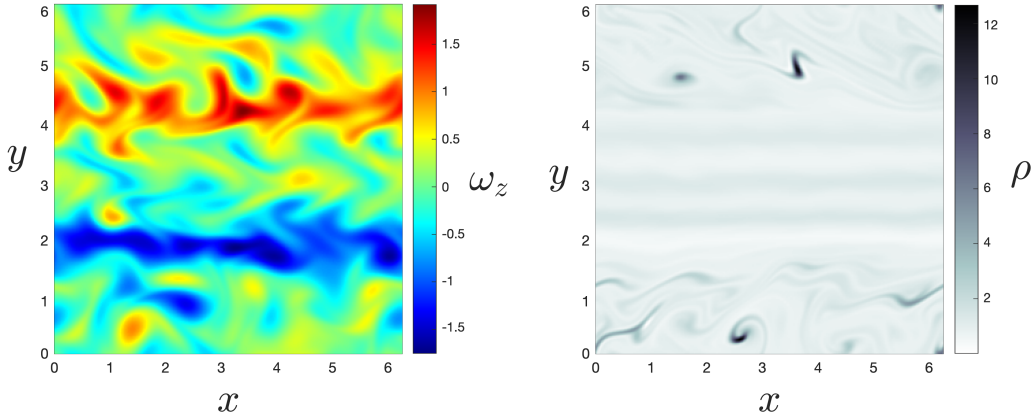
Figure 1: Snapshot of vorticity (left) and bubble density field (right) for a bimodal turbulent jet for $Re = 1000$, and bubble parameters $\epsilon = 0.05$ and $R = 2$.

4 Training of the closures

We study the effectiveness of the proposed closure scheme in two different setups. In the first setup, we take advantage of the translational invariance of the flow in the $x-$direction. This allows us to obtain a closed averaged equation for the $y-$profile of the jet. In the second case we do not rely on this symmetry and obtain closures directly for the two-dimensional flow. We will compare the adopted architectures for both the case of utilizing the energy constraint presented above and not.

4.1 One-dimensional closures for the jet profile

We take advantage of the translational invariance of the flow in the $x-$direction and select the spatial-averaging operator to be integration along the full x-direction and local spatial averaging along the y-direction:

$$\bar{f}(y) = \frac{1}{2\pi} \iint_{S^2} w_l(y' - y) f(x, y') \mathrm{d}x \mathrm{d}y' \tag{21}$$

where, $w_l(y' - y) = \frac{1}{2l}[H(y' - y - l) - H(y' - y + l)]$ is the piece-wise constant averaging window of length $2l$; here $l = \frac{2\pi}{20}$. Applying the averaging operator to the governing equations we obtain the equation for the $y-$profile of the jet. Note that based on our averaging operator for this case, we have $\bar{u}_2 = 0$ and $\bar{u}_1$ is only a function of y. To this end, we have:

$$\partial_t \bar{u}_1 = -\mathbb{D}_{u_1} + \frac{1}{\mathrm{Re}} \partial_y^2 \bar{u}_1 + \nu \nabla^{-4}(\bar{u}_1 - u_{1,jet}) + \overline{F}_1, \tag{22}$$

$$\partial_t \bar{\rho} + \partial_y (\overline{v_2}\, \bar{\rho}) + \mathbb{D}_\rho = \nu_2 \partial_y^4 \bar{\rho}. \tag{23}$$

Therefore, the objective function used for training of the flow closure takes the form:

$$L(\theta_1) = \int_{\Omega \times T} \left\| \mathbb{D}_{u_1}\left[\theta_1; \overline{\xi}\right] - \overline{(\mathbf{u}' \cdot \nabla)\mathbf{u}'} \right\|^2 \mathrm{d}\mathbf{x}\mathrm{d}t + \lambda \int_T \left| \int_\Omega \bar{u}_1 \cdot \mathbb{D}_{u_1}\left[\theta_1; \overline{\xi}\right] \mathrm{d}\mathbf{x} - A[\mathbf{u}] \right| \mathrm{d}t, \tag{24}$$

9

where $A[\mathbf{u}] = -\int_\Omega \overline{\mathbf{u}' \cdot (\mathbf{u}' \cdot \nabla)\mathbf{u}'}\mathrm{d}\mathbf{x}$. On the other hand, the objective function for the density field closure takes the form:

$$L(\theta_2) = \int_{\Omega \times T} \left|\left| \mathbb{D}_\rho \left[\theta_2; \overline{\zeta}[\alpha(\mathbf{x}), \chi(t)] \right] - \nabla \cdot \left(\overline{\mathbf{v}'\rho'} \right) \right|\right|^2 d\mathbf{x}dt. \tag{25}$$

The neighborhood $\alpha(y)$ is selected to have five nodes in total:

$$\alpha(y) = \{y + m\delta y\}, \ \delta y = 2\pi/80, \ \ m = -2, -1, ..., 2,$$

while the temporal horizon in the past is selected as

$$\chi(t) = \{t - m\delta\tau\}, \delta\tau = 10^{-2}, \ m = 1, 2, ..., 12.$$

Both the spatial extent of the neighborhood and the memory are chosen as the threshold values above which any further increase does not result significant difference in the training and validation errors.

4.1.1 Neural network architecture

We assess two different architectures for our closure scheme. In the first case, we represent the flow closure with 3 convolutional-LSTM layers and the density closure with 4 convolutional-LSTM layers (16 time-delays). Further increase of the number of layers does not offer any significant improvement in the training and validation errors. The adopted architectures are presented in Figure 2 (a-b). For the second machine learning architecture we use 4-layer temporal convolutions to model the memory terms of our closure for both the flow and the density fields. The architecture in this case is depicted in Figure 3(a-b). An important difference between the two architectures that is worth emphasizing is associated with their computational cost. Specifically, in the LSTM architecture we have a memory term that is updated at each time-step and to this end, LSTM needs to only operate on the flow features at each time-step. On the other hand, TCN layers operate on the entire included time-history making them more computationally expensive.

4.1.2 Feature selection

The selection of the flow features that are used as inputs for the data-driven closures is done numerically by testing different combinations of basic flow features. We eventually choose the combination that minimizes the training and the validation error. It is important to emphasize that if we rely only on the training error we run the risk of over-fitting. We carry out this process individually for each of the closure terms for the Navier-Stokes equation and the transport equation.

For the closure term $\mathbb{D}_\mathbf{u}$ we select as possible flow features the quantities: $\overline{u}_1, \partial_t \overline{u}_1, \partial_y(\overline{u'_1 u'_2})$. Table 1 summarizes performance for different combinations over 100 periods for the TCN and LSTM architectures with the physical constraint. We observe that the single most important feature is the ingredients of the material derivative, $D\overline{u}_1/Dt$. Also, we note that the best validation result is achieved by employing all the considered flow features.

Table 1: Feature selection for the one-dimensional closure of the Navier-Stokes.

ξ feature selection					
		cTCN		cLSTM	
Features	Dim	Train-MSE	Val-MSE	Train-MSE	Val-MSE
$\overline{u}_1$	1	0.094	1.712	0.102	1.501
$\partial_t\overline{u}_1$	1	0.037	0.535	0.041	0.501
$\partial_y(\overline{u'_1 u'_2})$	1	0.028	0.144	0.033	0.139
$\overline{u}_1, \partial_t\overline{u}_1$	2	0.042	0.418	0.056	0.511
$\overline{u}_1, \partial_y(\overline{u'_1 u'_2})$	2	0.023	0.159	0.028	0.157
$\partial_t\overline{u}_1, \partial_y(\overline{u'_1 u'_2})$	2	0.021	0.092	0.026	0.085
$\overline{u}_1, \partial_t\overline{u}_1, \partial_y(\overline{u'_1 u'_2})$	3	0.021	0.029	0.025	0.032

For the closure of the transport equation we carry out the same process in Table 2, where we present the training and validation error over 100 periods. We observe that the single most important feature is $\overline{\rho}$. Based on the mean-square error (both training and validation) we choose the combination of $\overline{u}, \overline{\rho}, \partial_t\overline{u}, \partial_y(\overline{\rho'v'})$. In Figure 2 we present the value of both validation and training error with respect to the number of epochs. These are quite similar, hinting towards generalizability of the predictions.

Table 2: Feature selection for closure of bubble transport equation.

ζ feature selection					
		cTCN		cLSTM	
Features	Dim	Train-MSE	Val-MSE	Train-MSE	Val-MSE
$\overline{\rho}$	1	0.109	0.150	0.123	0.171
$\overline{\boldsymbol{v}}$	2	0.603	0.673	0.592	0.625
$\overline{\boldsymbol{v}}, \overline{\rho}$	3	0.081	0.090	0.094	0.101
$\overline{\boldsymbol{v}}, \overline{\rho}, \partial_t\overline{\boldsymbol{v}}, \partial_t\overline{\rho}$	6	0.058	0.060	0.061	0.064
$\overline{\boldsymbol{v}}, \overline{\rho}, \partial_t\overline{\boldsymbol{v}}, \partial_y(\overline{\rho'v'_2})$	6	0.028	0.039	0.042	0.088
$\overline{\boldsymbol{v}}, \overline{\rho}, \partial_t\overline{\rho}, \partial_y(\overline{\rho'v'_2})$	5	0.027	0.036	0.035	0.049
$\overline{\boldsymbol{v}}, \overline{\rho}.\partial_t\overline{\boldsymbol{v}}, \partial_t\overline{\rho}, \partial_y(\overline{\rho'v'_2})$	7	0.025	0.031	0.033	0.044

Fluid NN architecture

$$\{\mathbb{D}_{u_1,0}, \mathbb{D}_{u_1,1}, \mathbb{D}_{u_1,2}, \ldots, \mathbb{D}_{u_1,t}\}$$

Smoothing

Convolutional LSTM

Convolutional LSTM

Convolutional LSTM

$$\{\overline{\xi}_0, \overline{\xi}_1, \overline{\xi}_2, \ldots, \overline{\xi}_t\}$$

(a)

Bubble NN architecture

$$\{\mathbb{D}_{\rho,0}, \mathbb{D}_{\rho,1}, \mathbb{D}_{\rho,2}, \ldots, \mathbb{D}_{\rho,t}\}$$

Smoothing

Convolutional LSTM

Convolutional LSTM

Convolutional LSTM

Convolutional LSTM

$$\{\overline{\xi}_0, \overline{\xi}_1, \overline{\xi}_2, \ldots, \overline{\xi}_t\}$$

(b)

Model loss

(c)

Model loss

(d)
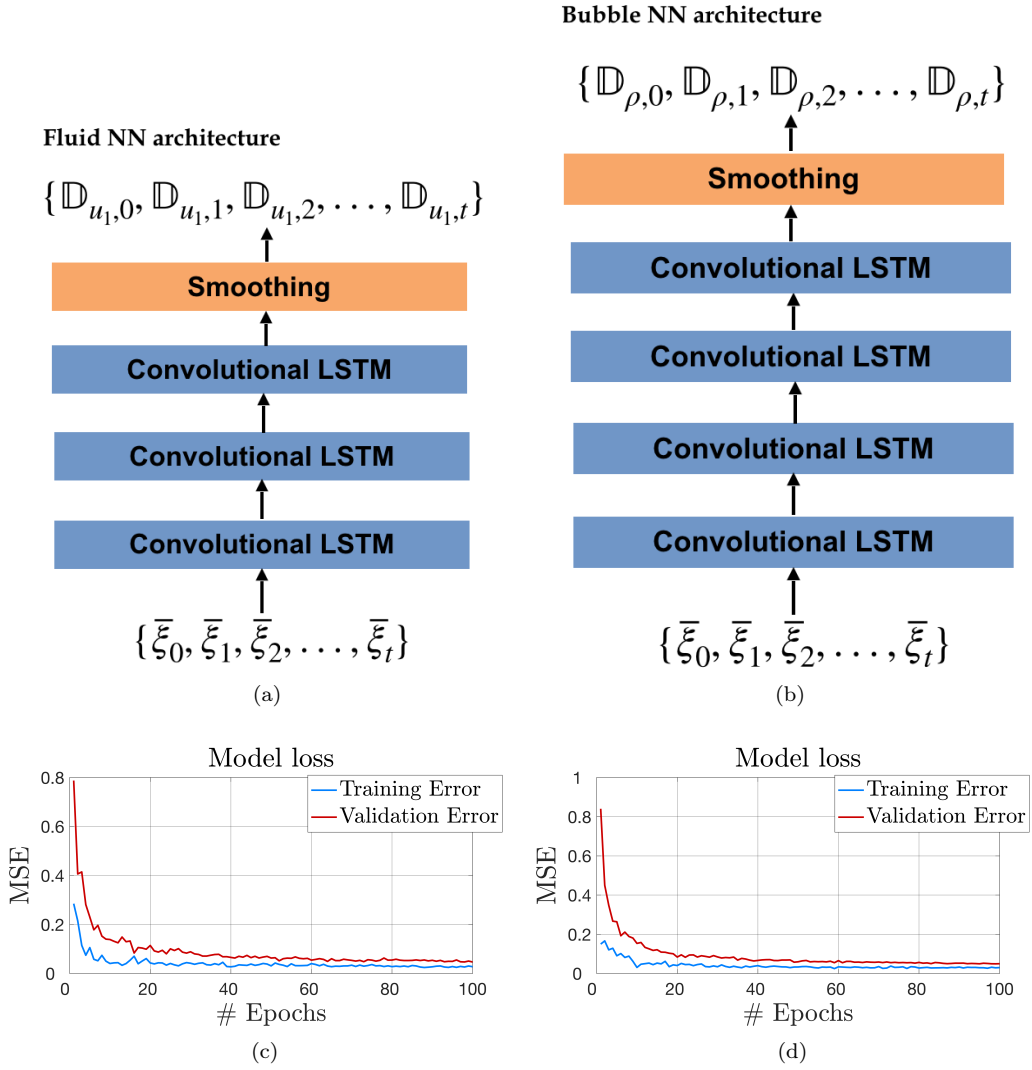
Figure 2: (a) Architecture of the LSTM neural network for parametrizing the term $\mathbb{D}_t = \partial_y(\overline{u_1' u_2'})$. (b) Architecture of the LSTM neural network for parametrizing the term $\mathbb{D}_t = \partial_y(\overline{v_2' \rho'})$. (c) Mean square training-error (solid line) and validation error (dashed line) for $\mathbb{D}_u$. (d) Mean square training-error (solid line) and validation error (dashed line) for $\mathbb{D}_\rho$.
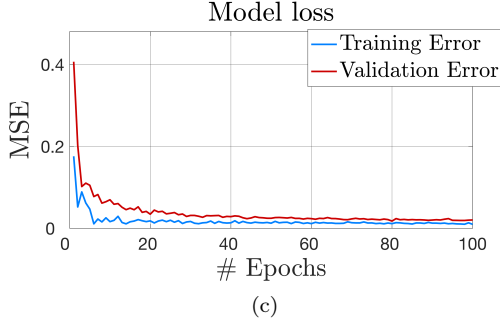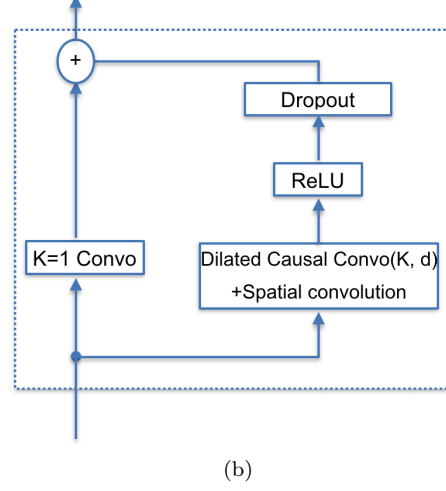
12

Fluid/Bubble NN architecture

$$\{\mathbb{D}_0, \mathbb{D}_1, \mathbb{D}_2, \ldots, \mathbb{D}_t\}$$

Smoothing

Residual block(K=2,d=8)

Residual block(K=2,d=4)

Residual block(K=2,d=2)

Residual block(K=2,d=1)

$$\{\overline{\overline{\xi}}_0, \overline{\overline{\xi}}_1, \overline{\overline{\xi}}_2, \ldots, \overline{\overline{\xi}}_t\}$$

(a)

+

Dropout

ReLU

K=1 Convo

Dilated Causal Convo(K, d)

+Spatial convolution

(b)

Model loss

— Training Error
— Validation Error

MSE

Epochs

(c)

Model loss

— Training Error
— Validation Error

MSE

Epochs

(d)
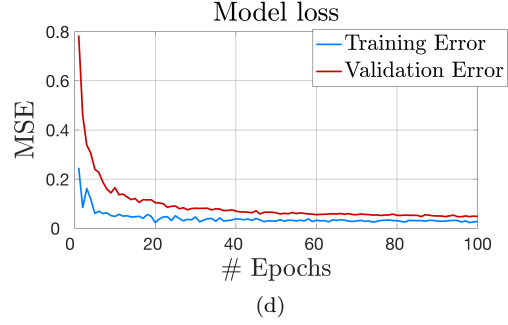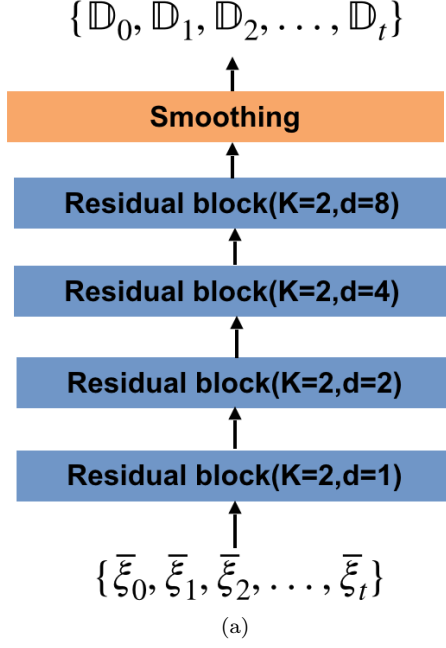
Figure 3: (a) Architecture of the TCN neural network for parametrizing the term $\partial_y(\overline{u_1' u_2'})$ and $\partial_y(\overline{v_2' \rho'})$. (b) Inner architecture of a residual block. (d) Mean square training-error (solid line) and validation error (dashed line) for $\mathbb{D}_u$. (e) Mean square training-error (solid line) and validation error (dashed line) for $\mathbb{D}_\rho$.

4.2 Two-dimensional closures

In this case our averaging operator was chosen to be local in both dimensions, x and y:

$$\bar{f}(x, y) = \frac{1}{2\pi} \iint_{S^2} w_{l_x}(y' - y) w_{l_y}(y' - y) f(x, y') dx dy', \tag{26}$$

where $l_x = l_y = \frac{2\pi}{12}$ are the averaging windows in the x and y directions respectively. As objective functions we used equations (15) and (16). The spatial and temporal neighborhood used in the

13

closures are chosen as:

$$\alpha(\mathbf{y}) = \{\mathbf{y} + m_1\delta\mathbf{x} + m_2\delta\mathbf{y}\}, \ \delta x = 2\pi/48, \delta y = 2\pi/48 \ \text{ and } \ m_1, m_2 = -2, -1, ..., 2,$$

$$\chi(t) = \{t - m\delta\tau\}, \delta\tau = 1/100, \ m = 1, 2, ..., 12.$$

Similarly with the one-dimensional closures, these numbers are based on the fact that further increase did not result significant difference in the training and validation errors. We employ the same neural network architectures that we used in the previous section.

4.2.1 Feature selection

For the closure term $\mathbb{D}_\mathbf{u}$ (corresponding to the fluid flow) we try as possible flow features the quantities $\overline{\mathbf{u}}, \partial_t\overline{\mathbf{u}}, \mathbb{D}_\mathbf{u}$. Results are shown for the case where the constraint is adopted (cTCN and cLSTM) in Table 3 in terms of training and validation errors. In this case, we observe that the single most important feature is the history of the Reynolds stresses. Furthermore, we see that the optimal combination consists of all the examined features.

For the closure of the transport equation we carry out the same process in Table 4. We observe that the single most important features seems to be $\overline{\rho}$, similarly with the one-dimensional case. For the results that follow we choose the combination $\overline{\mathbf{v}}, \overline{\rho}, \partial_t\overline{\mathbf{v}}, \partial_t\overline{\rho}, \mathbb{D}_\rho$, which results in the minimum validation and testing errors.

Table 3: Feature selection for closure of Navier-Stokes.

ξ feature selection					
		cTCN		cLSTM	
Features	Dim	Train-MSE	Val-MSE	Train-MSE	Val-MSE
$\overline{\mathbf{u}} = (\overline{u_1}, \overline{u_2})$	2	0.235	0.521	0.258	0.572
$\partial_t\overline{\mathbf{u}} = (\partial_t\overline{u_1}, \partial_t\overline{u_2})$	2	0.098	0.480	0.112	0.388
$\mathbb{D}_\mathbf{u}$	2	0.081	0.094	0.092	0.114
$\overline{\mathbf{u}}, \partial_t\overline{\mathbf{u}}$	4	0.069	0.485	0.100	0.522
$\overline{\mathbf{u}}, \mathbb{D}_\mathbf{u}$	4	0.048	0.082	0.067	0.094
$\partial_t\overline{\mathbf{u}}, \mathbb{D}_\mathbf{u}$	4	0.027	0.048	0.034	0.063
$\overline{\mathbf{u}}, \partial_t\overline{\mathbf{u}}, \mathbb{D}_\mathbf{u}$	6	0.020	0.039	0.027	0.055

Table 4: Feature selection for closure of bubble transport equation.

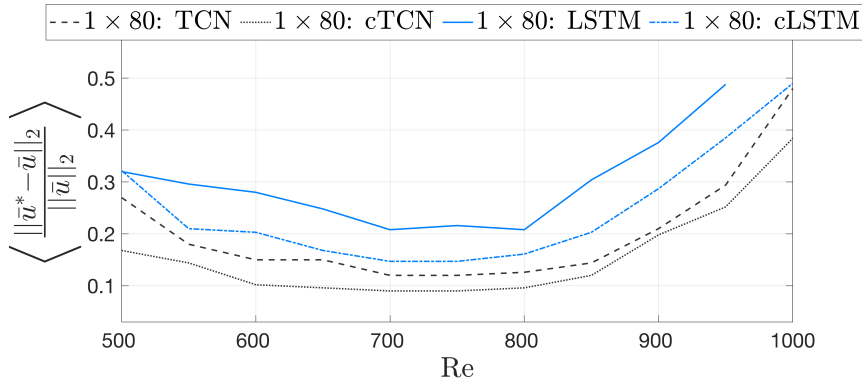| ζ feature selection					
		cTCN		cLSTM	
Features	Dim	Train-MSE	Val-MSE	Train-MSE	Val-MSE
$\overline{\rho}$	1	0.199	0.398	0.228	0.451
$\overline{\mathbf{v}} = (\overline{v_1}, \overline{v_2})$	2	0.320	0.591	0.318	0.515
$\overline{\mathbf{v}}, \overline{\rho}$	3	0.141	0.386	0.162	0.404
$\overline{\mathbf{v}}, \overline{\rho}, \partial_t\overline{\mathbf{v}}$	5	0.085	0.176	0.087	0.192
$\overline{\mathbf{v}}, \overline{\rho}, \partial_t\overline{\rho}, \mathbb{D}_\rho$	5	0.051	0.091	0.061	0.125
$\overline{\mathbf{v}}, \partial_t\overline{\rho}, \partial_t\overline{\mathbf{v}}, \mathbb{D}_\rho$	6	0.030	0.049	0.038	0.068
$\overline{\mathbf{v}}, \overline{\rho}, \partial_t\overline{\rho}, \partial_t\overline{\mathbf{v}}, \mathbb{D}_\rho$	7	0.016	0.032	0.031	0.047

Figure 4: Normalized error (28) for one-dimensional closure models using TCN, LSTM and and their constrained versions, cTCN and cLSTM for unimodal jets. Training data includes unimodal jets with $Re \in \{650, 750, 850\}$.

5 Validation and generalizability for one-dimensional closures

To showcase the generalizability properties of the obtained closures we train on unimodal jets and we test on bimodal ones. We mention again that the averaged model is one-dimensional and we use 80 points in space to simulate it. We compare the results of the averaged model with the predictions of the two-dimensional reference solutions that we computed using a spectral method and 256^2 modes. Each training case contains data in the time-interval $T = [200, 600]$.

5.1 Validation on unimodal jets

The unperturbed jet profile is chosen as,

$$u_{1,jet} = \exp\left[-2(y - \pi)^2\right], \quad u_{2,jet} = 0. \tag{27}$$

We train four different models on unimodal jets of $Re \in \{650, 750, 850\}$. We use LSTM and TCN architectures with and without enforcing the physical constraint of eq. (24). In Figure 4, we present the time- and $y-$ averaged mean-square error between the $x-$averaged profile of the DNS, $\bar{u}^*$ and the coarse scale model, $\bar{u}$:

$$||\bar{u}^* - \bar{u}||_2^2 = \frac{1}{2\pi T} \int_0^{2\pi} \int_{t_0}^{t_0+T} (\bar{u}^*(y, t) - \bar{u}(y, t))^2 dy dt. \tag{28}$$

We observe that the TCN models clearly outperform the LSTM based closures. Moreover, training using the objective function that includes the physical constraint (eq. (24)) for the advective terms (cTCN and cLSTM) improves significantly the testing results for the two architectures by 23% and 25%, respectively (Table 5). This improvement comes at no additional cost in terms of data, but only using the physical constraint associated with the advection terms, which does not depend on the knowledge of any physical quantity of the flow or any other system-specific information.

In Figure 5 we present additional results for the cTCN model (best performer). We showcase results both for the time-averaged profile for the fluid velocity and for the bubble distribution for $Re = 1000$. Comparisons are made between the time-averaged results that the one-dimensional
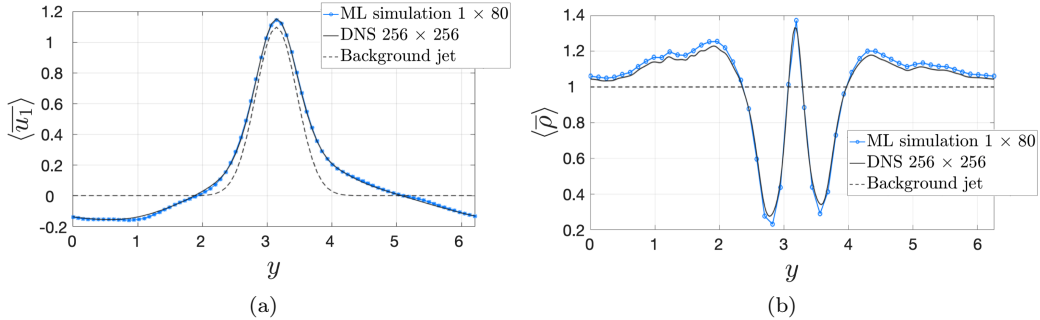
15

Figure 5: Time-averaged profile of $\overline{u}_1$ (left) and $\overline{\rho}$ (right) for the one-dimensional cTCN closure model (blue line) and the DNS (black line). The shape of the jet that is imposed by the large-scale forcing is depicted with dashed line. The simulation corresponds to $Re = 1000$ while training included $Re \in \{650, 750, 850\}$.

closure scheme produces and the time- and $x-$ averaged results of the two-dimensional reference solution. Specifically, the time-averaged jet-profile is computed as

$$\langle \overline{u}_1 \rangle(y) = \frac{1}{T} \int_{t_0}^{t_0+T} \overline{u}_1(y,t) dt, \qquad (29)$$

where $\overline{u}_1$ is the $x-$averaged reference solution and the temporal averaging parameters are chosen as $t_0 = 200$ and $T = 400$; note that we omit the first transient part of the simulation.

The results appear to be in very good agreement showcasing that our closure scheme is able to predict the statistical steady state of the flow. We can also observe the Rayleigh instability that the initial jet profile (dashed line) undergoes due to the excitation by the external forcing. We note that the slight asymmetry that the velocity profile exhibits is due to a minor inhomogeneity of the forcing term along the y direction. We apply the same operation described above to compute $\langle \overline{\rho} \rangle$ from both the reference solution and the data-based closure scheme (Figure 5(b)). Again, we obtain very good agreement between the machine learning approach and the reference solution, which has a non-trivial form, as the bubbles seem to cluster around the core of the jet and be repelled from the adjacent areas of the jet core.

5.2 Testing generalizability on bimodal jets

Next we test the generalizability of the closure schemes presented in the previous section on bimodal jets. Once again we state that we train on unimodal jets (as described previously) while we test our scheme on bimodal jets with the unperturbed jet-structure of the fluid flow chosen as

$$u_1 = \exp[-3(y - 0.8\pi)^2] + \exp[-3(y - 1.2\pi)^2], \quad Re \in [500, 1000]. \qquad (30)$$

In Figure 6 we present the normalized mean-square error (28) between the reference solution and the one predicted by the closure model. As we can see, employing the physical constraint during training significantly improves the results. Specifically, we note that the errors of the cTCN and cLSTM remain to the same levels observed in the unimodal jet case (i.e. we have good generalizability properties in different flows), while the unconstrained version has significantly worse performance compared with the unimodal jet case. In addition, we observe much better behavior of the constrained
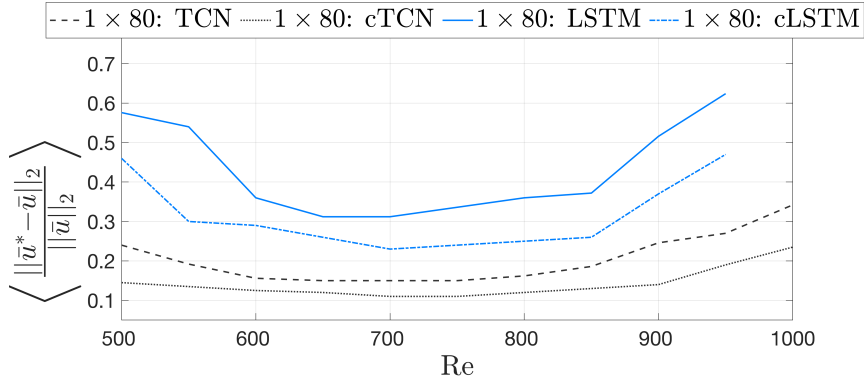
16

Figure 6: Normalized error (28) for one-dimensional closure models error for one-dimensional closures applied on bimodal jets. Training data includes unimodal jets with $Re = \{650, 750, 850\}$.

closures when we move to Reynolds numbers higher than the ones used for training. In fact, for sufficiently large Reynolds the schemes based on the unconstrained closures become unstable.

Results regarding the time-averaged jet profile of $\overline{u}_1$ and ρ are depicted in Figure 7 for the cTCN architecture, where we can observe the excellent agreement of the predicted profile with DNS. We also apply the closure scheme on the transport equation (7) to compute the distribution of bubbles. We present the comparison of the mean distribution of bubbles between the cTCN closure model and DNS in Figure 7(b). We note that the error for this case is slightly more pronounced compared with the one observed for the mean flow velocity in Figure 7(a). This can be attributed to two factors: i) the predictions of the transport model rely on the predictions of the coarse-scale model for the velocity field (hence error accumulates); and ii) the closure model for the bubbles relies only on data since the energy-preserving constraint is not relevant.
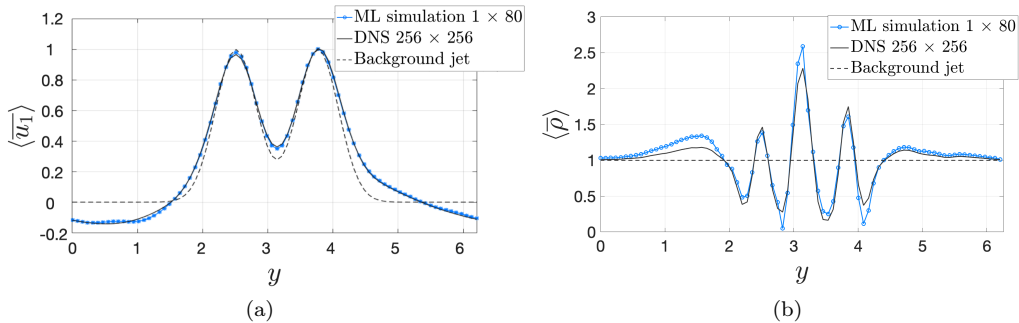

Figure 7: Time-averaged profile of $\overline{u}_1$ and $\bar{\rho}$ for the one-dimensional cTCN closure model (blue line) and the DNS (black line). The simulation corresponds to $Re = 1000$ and a bi-model background jet, while training data for the closures correspond to unimodal jets with $Re \in \{650, 750, 850\}$. The shape of the jet that is imposed by the large-scale forcing is depicted with dashed line.

A summary of the error improvement in the one-dimensional predictions due to the adoption of the physical constraint is presented in Table 5. We present the improvement of the mean-square error for the mean flow, averaged over different Reynolds numbers. For all cases this percentage

ranges between 23% and 29% with the improvement being more pronounced for the bimodal setups, i.e. the setup that was not used for training.

Table 5: Error decrease (Reynolds-averaged) due to the physical constraint for one-dimensional closure schemes.

Architecture	Jet-type	Error decrease
TCN-1D	Unimodal	23%
LSTM-1D	Unimodal	25%
TCN-1D	Bimodal	29%
LSTM-1D	Bimodal	27%

6 Validation and generalizability for two-dimensional closures

Here we aim to showcase the application of our method on two-dimensional coarse-scale closures. As previously, we consider two cases: (i) we train on *unimodal* jets for flows with Reynolds number $Re \in \{650, 750, 850\}$ and test on *unimodal* jets in the range $Re \in [500, 1000]$ and not included in the training set; (ii) we once again train on *unimodal* jets (same Reynolds as before) and test on *bimodal* jets in the same Reynolds range as in case (i). For the coarse-scale model we employ a resolution of 48×48, complemented with the ML-closure terms. We compare the energy spectra at the statistical steady state of the flows between the coarse-scale predictions and the two-dimensional reference solutions, i.e. for $t \in [200, 600]$.

6.1 Testing on a unimodal jet

In Figure 8 we present the space-time-averaged mean-square error between the $x - y$ locally averaged DNS flow field (using eq. (26)), $\bar{u}^*$, and the coarse scale model, $\bar{u}$:

$$||\bar{u}^* - \bar{u}||_2^2 = \frac{1}{(2\pi)^2 T} \int_0^{2\pi} \int_0^{2\pi} \int_{t_0}^{t_0+T} (\bar{u}^*(x,y,t) - \bar{u}(x,y,t))^2 dx dy dt. \tag{31}$$

The results are in full consistency with the one-dimensional closures, i.e. cTCN has the best performance. We also present a detailed comparison for $Re = 800$ between the coarse-model and the DNS simulation in terms of the energy spectrum and mean profile the flow (Figure 9). The energy spectrum is computed by obtaining the spatial Fourier transform at each time instant and then considering the variance of each Fourier coefficient over time. We plot the energy spectrum in terms of the absolute wavenumber values. For both the flow field and bubble field the coarse-model is able to accurately capture the mean profiles, as well as the large scale features of the spectrum.
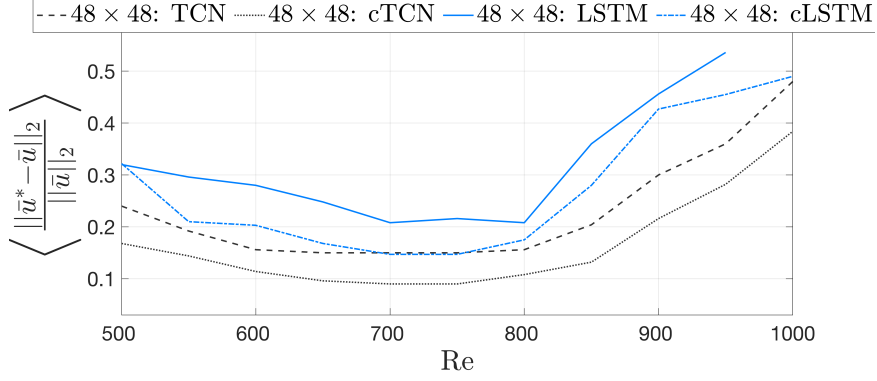
18

Figure 8: Normalized mean-square error (31) of each two-dimensional closure model using TCN, LSTM and their constrained versions for unimodal jets. Training includes data for unimodal jets with $Re \in \{650, 750, 850\}$.
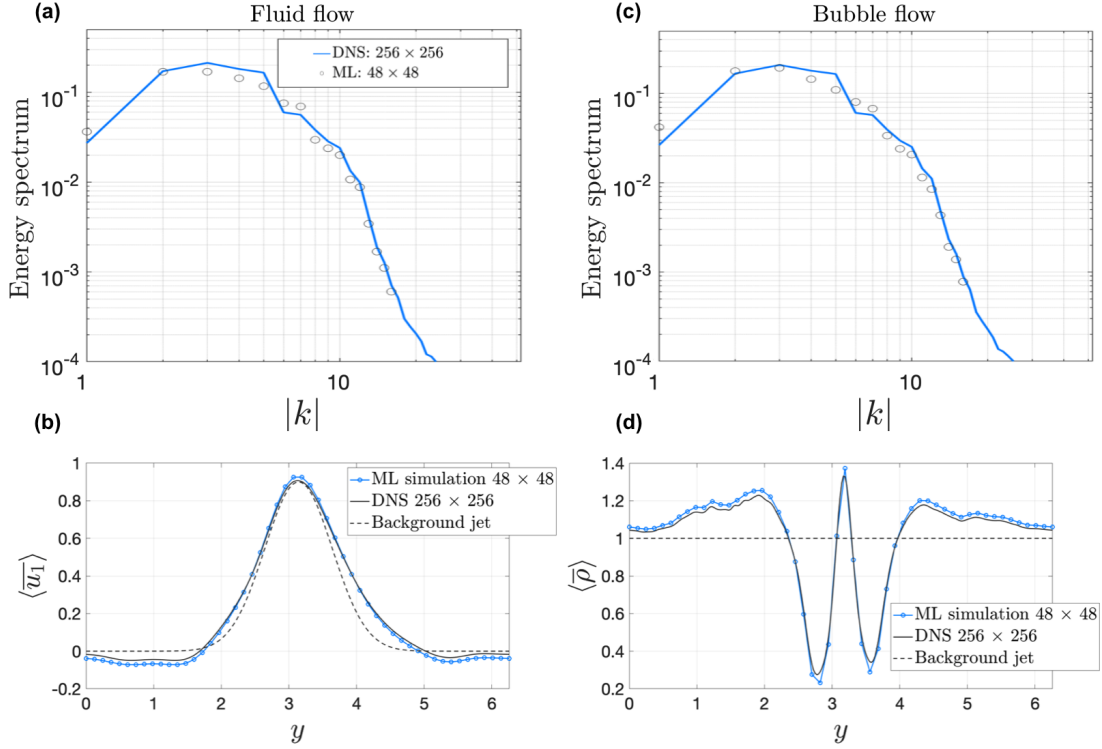

Figure 9: Comparison of energy spectra and mean profiles for the flow velocity field (a, c) and the bubble velocity field (b, d) for unimodal jets. Blue lines correspond to DNS simulations on a 256×256 grid and the black circles correspond to the coarse model using two-dimensional closures (cTCN) on a 48×48 grid for $Re = 800$.

19

6.2 Testing generalizability on bimodal-jets

We proceed to test the generalizability of the two-dimensional closure schemes on bimodal jets. The setup is identical with the one adopted for one-dimensional closures. In Figure 10 we compare the noormalized mean-square error between the locally averaged DNS solution and the one obtained form the coarse model. Consistently with the previous results the cTCN has the best performance. It is interesting to note that the performance is even better in the Reynolds regime outside the training data, i.e. for $Re > 850$.

The energy spectrum of the fluid velocity and the bubble velocity field, as well as the corresponding mean profiles are compared with DNS for $Re = 800$ in Figure 11. In this case, we note that while there is good agreement between the mean profiles, there is some discrepancy between the approximate and exact spectra. To understand better the source of this discrepancy we plot the energy spectrum of the flow in the k_x, k_y space (Figure 12). As it can be seen the coarse model overestimates the spread of the energy of the fluctuations only in the $x-$direction, which is consistent with the fact that the mean $y-$profile of the flow is accurately modeled. This is not surprising given that the developed closures in this paper are designed to capture well the mean flow characteristics and not necessarily the energy spectrum. A closure approach based on second-order statistical equations (see e.g. [32]) is beyond the scope of this work and will be considered elsewhere.
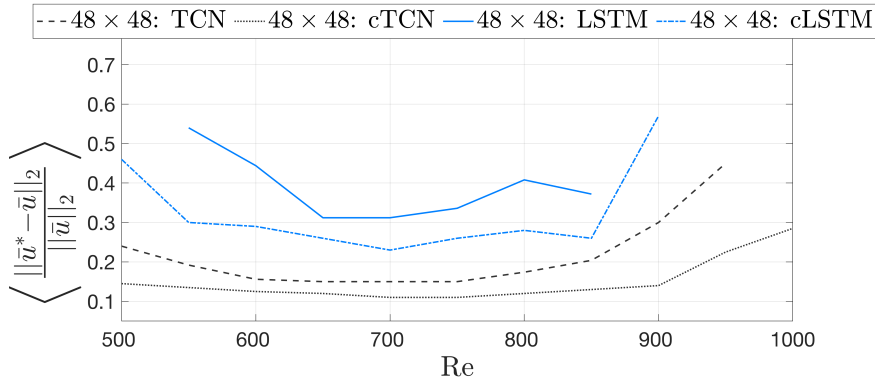

Figure 10: Normalized mean-square error (31) for two-dimensional coarse models applied on bimodal jet flows. Training used data from unimodal flows with $Re = \{650, 750, 850\}$.
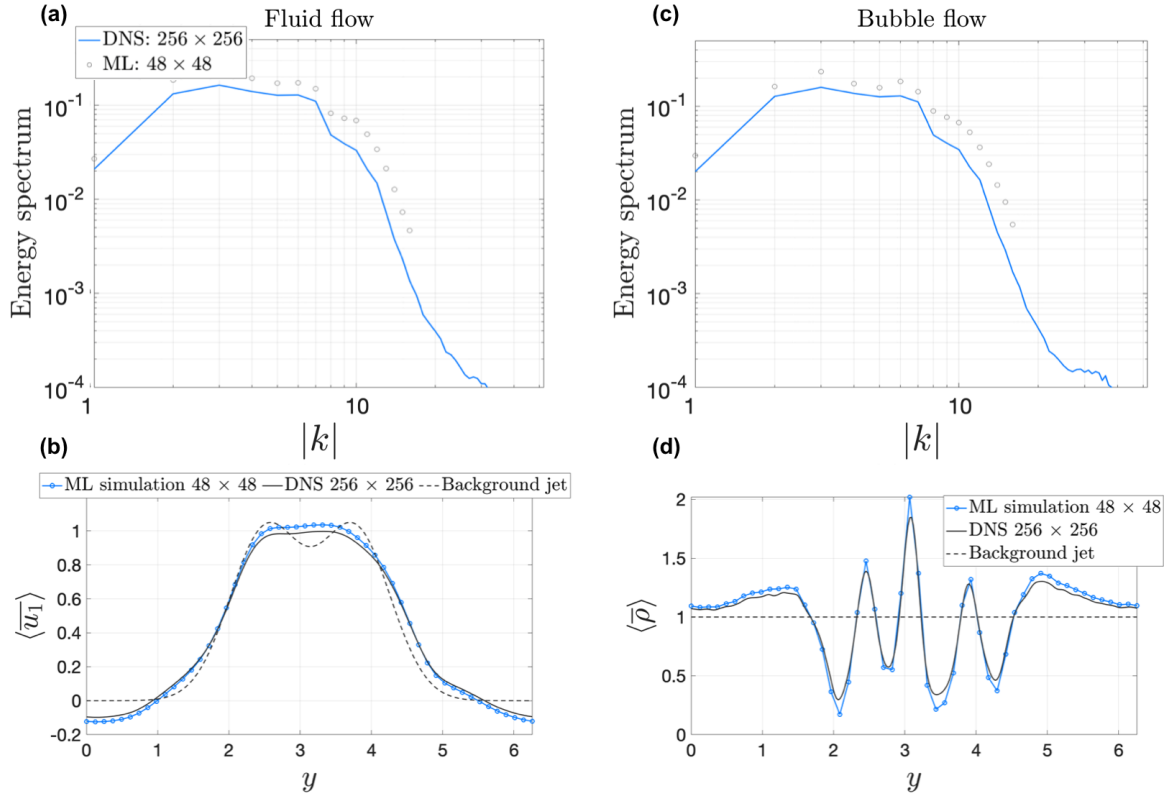
Figure 11: Comparison of energy spectra and mean profiles for the fluid velocity field (a, c) and the bubble velocity field (b, d) for the case of a bimodal jet with $Re = 800$. Blue lines correspond to DNS simulations on a 256×256 grid and the black circles correspond to a coarse-model with two-dimensional cTCN closures on a 48×48 grid.
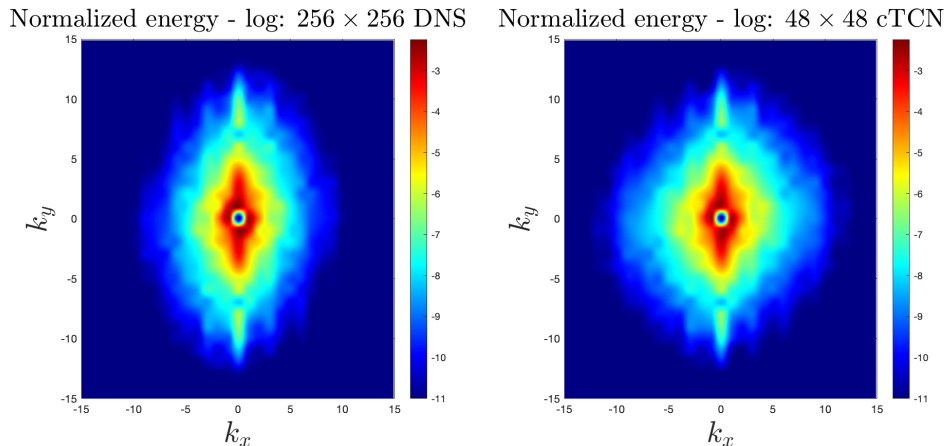
Figure 12: Energy spectrum of the fluid flow for the bimodal jet for $Re = 800$. Comparison between the DNS simulation (left) and the coarse-model based on 2D cTCN closures (right).

The overall improvement in the two-dimensional predictions due to the adoption of the physical constraint is summarized in Table 6, where we show the improvement of the mean-square error for the mean flow, averaged over different Reynolds numbers. We note that for the TCN architecture the improvement is more pronounced, close to 30%, and it is also robust for the case of a bimodal jet.

Table 6: Error decrease (Reynolds-averaged) due to the physical constraint for two-dimensional closure schemes.

Architecture	Jet-type	Error decrease
TCN-2D	Unimodal	30%
LSTM-2D	Unimodal	20%
TCN-2D	Bimodal	33%
LSTM-2D	Bimodal	31%

6.3   Dependence on the coarse-model grid-size

Finally, we showcase a numerical investigation for the relationship between the chosen grid-size for the coarse-scale simulations and the mean-square error of the velocity of the fluid flow. For all the results presented below, training data was chosen as previously (unimodal jets) and results are presented for bimodal jets. In Figure 13(a) we vary the size of a $N_x \times N_y$ grid with $N_x = N_y \in \{16, 24, 32, 40, 48, 64, 80, 96\}$. We notice that there is significant improvement in our predictions as we refine the grid up to a grid-size of 64×64 where the error saturates.

Since the variation of the mean profile of the flow is only along the y-direction, having a coarser resolution along the less significant x-direction should not hinder the predictions. To validate this property we maintain a constant discretization in the $x-$ direction and vary the grid-size only in the $y-$ direction. Results are demonstrated in Figure 13(b), showing clearly that by having a fine resolution only in the $y-$direction is sufficient to achieve comparable performance with the fine resolution case in both directions: $N_x = N_y = 96$.
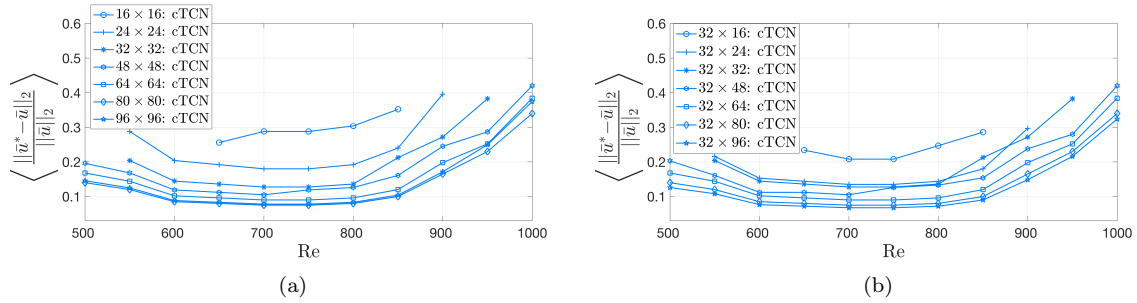
Figure 13: Root-mean-square error for different grid-sizes using two-dimensional cTCN closures on bimodal jet.

7 Conclusions

We have demonstrated the application of the energy conservation property of the advection terms on machine learning non-local closures for turbulent fluxes. We have adopted two neural network architectures, based on LSTM and TCN, to further include memory effects in our analysis. Clearly, the physical constraint is not restricted to these two frameworks and can be employed in other machine learning architectures, as well as other fluid systems (e.g. environmental flows). We demonstrated the computed closures in two-dimensional jets in an unstable regime and showed that closures obtained from unimodal jets can be used for different jet geometries. The adoption of the physical constraint significantly and consistently improved the accuracy of the mean-flow predictions obtained from the corresponding coarse-scale models independently of the adopted architecture or the flow setup. This improvement was in average 26% for one-dimensional closures and 29% for two-dimensional closures, being notably larger for flows that were not used for training. Moreover, the constraint improved the numerical stability of the coarse-scale models especially in Reynolds numbers, which were higher than the ones included in the training data sets. While the adopted examples are relatively simple, yet unstable, fluid flows, the presented energy constraints do not depend on the complexity of the flow. They are applicable for more complicated setups including boundary flows and transition phenomena, directions that we plan to pursue in the future.

References

[1] S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[2] Y. Bengio, P. Simard, P. Frasconi, et al. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[3] G. Boffetta and S. Musacchio. Evidence for the double cascade scenario in two-dimensional turbulence. *Physical Review E*, 82(1):016307, 2010.

[4] V. Borue. Spectral exponents of enstrophy cascade in stationary two-dimensional homogeneous turbulence. *Physical review letters*, 71(24):3967, 1993.

[5] A. Bracco and J. C. McWilliams. Reynolds-number dependency in homogeneous, stationary two-dimensional turbulence. *Journal of Fluid Mechanics*, 646:517–526, 2010.

[6] S. Brunton, B. Noack, and P. Koumoutsakos. Machine Learning for Fluid Mechanics. *Ann. Rev. Fluid Mech.*, 52:477–508, 2020.

[7] S. H. Bryngelson, A. Charalampopoulos, T. P. Sapsis, and T. Colonius. A gaussian moment method and its augmentation via lstm recurrent neural networks for the statistics of cavitating bubble populations. *International Journal of Multiphase Flow*, page 103262, 2020.

[8] H. Chen. Approximations of Continuous Functionals by Neural Networks with Application to Dynamic Systems. *IEEE Transactions on Neural Networks*, 1993.

[9] T. Chen and H. Chen. Universal Approximation to Nonlinear Operators by Neural Networks with Arbitrary Activation Functions and Its Application to Dynamical Systems. *IEEE Transactions on Neural Networks*, 1995.

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

[11] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[12] G. Haller and T. Sapsis. Where do inertial particles go in fluid flows? *Physica D: Nonlinear Phenomena*, 237(5):573–583, 2008.

[13] P. E. Hamlington and W. J. Dahm. Reynolds stress closure including nonlocal and nonequilibrium effects in turbulent flows. In *39th AIAA Fluid Dynamics Conference*, 2009.

[14] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[15] J. Ling, R. Jones, and J. Templeton. Machine learning strategies for systems with invariance properties. *Journal of Computational Physics*, 318:22–35, 2016.

[16] J. Ling, A. Kurzawski, and J. Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.

[17] M. Ma, J. Lu, and G. Tryggvason. Using statistical learning to close two-fluid multiphase flow equations for a simple bubbly system. *Physics of Fluids*, 27(9):092101, 2015.

[18] M. Ma, J. Lu, and G. Tryggvason. Using statistical learning to close two-fluid multiphase flow equations for bubbly flows in vertical channels. *International Journal of Multiphase Flow*, 85:336–347, 2016.

[19] A. J. Majda. Statistical energy conservation principle for inhomogeneous turbulent dynamical systems. *Proceedings of the National Academy of Sciences*, 112(29):8937–8941, jul 2015.

[20] R. Maulik, O. San, A. Rasheed, and P. Vedula. Subgrid modelling for two-dimensional turbulence using neural networks. *Journal of Fluid Mechanics*, 858:122–144, 2019.

[21] A. Mazzino, P. Muratore-Ginanneschi, and S. Musacchio. Scaling properties of the two-dimensional randomly stirred navier-stokes equation. *Physical review letters*, 99(14):144502, 2007.

[22] M. Milano and P. Koumoutsakos. Neural network modeling for near wall turbulent flow. *Journal of Computational Physics*, 182(1):1–26, 2002.

[23] P. Moin, K. Squires, W. Cabot, and S. Lee. A dynamic subgrid-scale model for compressible turbulence and scalar transport. *Physics of Fluids A*, 1991.

[24] G. Novati, H. L. de Laroussilhe, and P. Koumoutsakos. Automating turbulence modelling by multi-agent reinforcement learning. *Nature Machine Intelligence*, 3:87–96, 2021.

[25] F. Porté-Agel, C. Meneveau, and M. Parlange. A scale-dependent dynamic model for large-eddy simulation: Application to a neutral atmospheric boundary layer. *Journal of Fluid Mechanics*, 2000.

[26] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 2019.

[27] M. Raissi, Z. Wang, M. S. Triantafyllou, and G. E. Karniadakis. Deep learning of vortex-induced vibrations. *Journal of Fluid Mechanics*, 2019.

[28] M. Raissi, A. Yazdani, and G. E. Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 2020.

[29] P. Sagaut and Y.-T. Lee. Large Eddy Simulation for Incompressible Flows: An Introduction. Scientific Computation Series. *Applied Mechanics Reviews*, 2002.

[30] M. Samiee, A. Akhavan-Safaei, and M. Zayernouri. A fractional subgrid-scale model for turbulent flows: Theoretical formulation and a priori study. *Physics of Fluids*, 32(5), 2020.

[31] T. P. Sapsis. Attractor local dimensionality, nonlinear energy transfers, and finite-time instabilities in unstable dynamical systems with applications to 2D fluid flows. *Proceedings of the Royal Society A*, 469(2153):20120550, 2013.

[32] T. P. Sapsis and A. J. Majda. A statistically accurate modified quasilinear Gaussian closure for uncertainty quantification in turbulent dynamical systems. *Physica D*, 252:34–45, 2013.

[33] T. P. Sapsis and A. J. Majda. Statistically accurate low-order models for uncertainty quantification in turbulent dynamical systems. *Proceedings of the National Academy of Sciences of the United States of America*, 110(34), 2013.

[34] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[35] A. P. Singh, S. Medida, and K. Duraisamy. Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils. *AIAA Journal*, pages 2215–2227, 2017.

[36] K. Stengel, A. Glaws, D. Hettinger, and R. N. King. Adversarial super-resolution of climatological wind and solar data. *Proceedings of the National Academy of Sciences of the United States of America*, 117(29):16805–16815, 2020.

[37] F. Takens. Detecting strange attractors in turbulence. In *Dynamical systems and turbulence, Warwick 1980*, pages 366–381. Springer, 1981.

[38] A. Venkatraman, M. Hebert, and J. A. Bagnell. Improving multi-step prediction of learned time series models. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[39] P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos. Data-driven forecasting of high-dimensional chaotic systems with long-short term memory networks. *Proceedings of the Royal Society A*, 474(2213):20170844, 2018.

[40] P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos. Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2213):20170844, 2018.

[41] P. R. Vlachas, J. Pathak, B. R. Hunt, T. P. Sapsis, M. Girvan, E. Ott, and P. Koumoutsakos. Backpropagation algorithms and Reservoir Computing in Recurrent Neural Networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks*, 126:191–217, 2020.

[42] Z. Y. Wan and T. P. Sapsis. Machine learning the kinematics of spherical particles in fluid flows. *Journal of Fluid Mechanics*, 857:1–11, 2018.

[43] Z. Y. Wan, P. Vlachas, P. Koumoutsakos, and T. Sapsis. Data-assisted reduced-order modeling of extreme events in complex dynamical systems. *PloS one*, 13(5):e0197704, 2018.

[44] Z. Y. Wan, P. R. Vlachas, P. Koumoutsakos, and T. P. Sapsis. Data-assisted reduced-order modeling of extreme events in complex dynamical systems. *PLoS ONE*, 13, 2018.

[45] J.-L. Wu, H. Xiao, and E. Paterson. Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. *Physical Review Fluids*, 3(7):074602, 2018.

[46] X. I. Yang, S. Zafar, J. X. Wang, and H. Xiao. Predictive large-eddy-simulation wall modeling via physics-informed neural networks. *Physical Review Fluids*, 2019.

[47] J. Zhuang, D. Kochkov, Y. Bar-Sinai, M. P. Brenner, and S. Hoyer. Learned discretizations for passive scalar advection in a 2-D turbulent flow. apr 2020.