
A POSTERIORI CLOSURE OF TURBULENCE MODELS: ARE SYMMETRIES PRESERVED?

André Freitas^{*1,2}, Kiwon Um², Mathieu Desbrun³, Michele Buzzicotti¹, Luca Biferale¹

¹Dept. Physics and INFN, University of Rome "Tor Vergata" and INFN, Italy

²LTCI, Télécom Paris, IP Paris, France

³LIX, Inria and École Polytechnique, IP Paris, France

*Corresponding author: andre.freitas@roma2.infn.it

March 11, 2026

ABSTRACT

Turbulence modeling remains a longstanding challenge in fluid dynamics. Recent advances in data-driven methods have led to a surge of novel approaches aimed at addressing this problem. This work builds upon our recent work [Phys. Rev. Fluids 10, 044602 (2025)], where we introduced a new closure for a shell model of turbulence using an a posteriori (or solver-in-the-loop) approach. Unlike most deep learning-based models, our method explicitly incorporates physical equations into the neural network framework, ensuring that the closure remains constrained by the underlying physics benefiting from enhanced stability and generalizability. In this paper, we further analyze the learned closure, probing its capabilities and limitations. In particular, we look at joint probability density functions between resolved and unresolved variables, as well as the scale invariance of multipliers (ratios between adjacent shells) within the inertial range. Although our model excels in reproducing high-order statistical moments, it breaks this known symmetry near the cutoff, indicating a fundamental limitation. We discuss the implications of these findings for subgrid-scale modeling in 3D turbulence and outline directions for future research.

1 Introduction

Turbulence is ubiquitous in nature, from the coffee we stir in the morning to the primordial states of our universe [1]. Despite its ubiquity, understanding and modeling turbulence remains one of the most challenging tasks in classical physics. While we have known its governing equations for over a century [2], the full phenomenology of turbulence remains elusive [3, 4]. Simulating turbulence numerically is also extremely computationally demanding [5, 6]. The degrees of freedom scale as a power law of the Reynolds number, $Re = UL/\nu$ (where U is a characteristic velocity, L the integral length scale, and ν the kinematic viscosity), which means that studying very high Reynolds number turbulence remains impossible for the most part. One approach to reduce the degrees of freedom is Large Eddy Simulation (LES) [7], where a filter is applied, typically within the inertial range, to resolve only the largest, most energy-containing eddies. However, to close the LES system, the effects of unresolved subgrid scales must be modeled. This constitutes the LES modeling problem. This problem is often addressed phenomenologically by introducing an eddy diffusivity term to stabilize the system [8]. This solution, albeit useful for practical application, fails to capture the correct subgrid scale statistics. Moreover, although decades of research have been spent in LES modeling [8–14], there are still fundamental open questions in the field. Can a subgrid-scale closure be formulated to accurately capture the statistics of the resolved scales, especially in high Reynolds number flows where intermittency [15] plays a crucial role? In turbulence, velocity fields are not statistically self-similar due to intermittency, making it essential for closures to account for the multifractal nature of the subgrid scales [16].

Recently, deep learning (DL) [17] approaches have gained attention for addressing the LES modeling problem [13]. DL excels at approximating functions with unknown functional forms [18, 19]. By learning from data, DL methods can, albeit in a black-box manner [20], accurately approximate high-dimensional functions [21]. To investigate the

applicability of DL in this context, we use shell models of turbulence [22] as our playground. Shell models offer a simplified representation of 3D homogeneous isotropic turbulence (HIT), providing a tractable alternative to the full Navier-Stokes equations. Despite their simplicity, shell models preserve key turbulence phenomena, such as energy cascade, intermittency, anomalous scaling, and multifractal energy dissipation.

Our primary focus is on the ability of subgrid-scale (SGS) models to accurately capture extreme events and rare fluctuations, which demand high precision, extensive statistical sampling, and long inertial ranges. Learning SGS models from data requires long temporal evolutions, presenting a fundamental challenge for LES of three-dimensional turbulence. This challenge is currently best addressed through shell models. Although LES of three-dimensional HIT remains the ideal test case, existing machine learning-based LES tools are hindered by their low resolution and the proximity of the cutoff to the forcing scales, preventing the emergence of anomalous behavior. Shell models, in contrast, provide a more suitable framework for rigorously testing the effectiveness of LES models under conditions of strong intermittency and for assessing whether DL approaches can improve the tracking of extreme fluctuations over time. Furthermore, higher-order statistics—particularly of fourth order and beyond—naturally involve non-local interactions in Fourier space, implying that SGS fluctuations can influence the resolved scales in non-trivial ways. This observation is critical when aiming for accuracy beyond second-order moments, which are typically the focus of LES validation. Shell models thus offer a controlled environment to probe whether a closure model can correctly capture such high-order interactions.

Several previous studies have explored closures within the shell model framework. Biferale et al. [12] initially approached the closure problem phenomenologically, using Kolmogorov multipliers. More recently, Ortali et al. [23] employed deep learning to learn the closure, while Domingues Lemos [24, 25] integrated probabilistic machine learning tools within the hidden symmetric framework of shell models [26] to address the closure. Building on these developments, we have also contributed to the field [27] through a solver-in-the-loop approach [28] based on the differentiable physics paradigm. This method unrolls the system in time during training, allowing the neural network to interact with a numerical solver for the governing equations over an arbitrary number of time steps before the loss function is computed. Rather than attempting to match the instantaneous, *a priori* evolution of the ground truth, we adopt an *a posteriori* approach, where the network learns how its errors propagate through time while receiving more realistic inputs. The induced closure was analyzed by examining high-order moments such as flatness up to the tenth order, demonstrating good agreement with the ground truth and unconditionally stable behavior.

This paper further extends our previous investigation by studying the learned closure in greater detail. While our earlier work [27] showed that the model reproduces the statistics of the resolved variables with high accuracy, here we test a stricter criterion: whether it can also recover nontrivial correlations between resolved and unresolved variables. To this end, we analyze joint probability density functions and other multi-scale statistics. We also investigate Kolmogorov multipliers [29], which are known to exhibit universal, scale-invariant statistics within the inertial range. These multipliers, especially those near the cutoff, intrinsically couple resolved and unresolved variables and therefore provide a sensitive diagnostic of whether the learned closure preserves or breaks this symmetry.

The remainder of the paper is structured as follows. In Section 2, we introduce the shell model framework and the mathematical formulation of our problem. Section 3 details the solver-in-the-loop approach and the methodology for learning an *a posteriori* closure. In Section 4, we present our results, focusing on statistical properties and preservation of known symmetries. Finally, we discuss our findings in Section 5 and outline potential directions for future research.

2 Shell models

Shell models are a class of turbulence models. They model homogeneous isotropic turbulence in Fourier space, considering only sparse coupling between the different shells. We discretize the 1D Fourier space with a geometric progression of shells that are equispaced on a logarithmic lattice, $k_n = k_0 \lambda^n$, where one typically uses $\lambda = 2$ and $k_0 = 1$. We consider the Sabra shell model [30], where the set of ordinary differential equations governing the dynamics of the different shells is given by:

$$\frac{du_n}{dt} = i \left(ak_{n+1}u_{n+2}u_{n+1}^* + bk_nu_{n+1}u_{n-1}^* - ck_{n-1}u_{n-1}u_{n-2} \right) - \nu k_n^2 u_n + f_n, \quad (1)$$

where $u_n \in \mathbb{C}$, represents the velocity fluctuations at a given shell n and $*$ denotes complex conjugation. We set the free parameters $(a, b, c) = (1, -1/2, -1/2)$ such that $a + b + c = 0$, which corresponds to conservation of energy $E = \sum_n |u_n|^2$ and helicity $H = \sum_n (a/c)^n |u_n|^2$ in the inviscid, unforced limit.

The first term in the RHS of Equation 1 represents the non-linear term, where we can see that only nearest and next-to-nearest neighbour interactions are preserved, making this model local in Fourier space; the second term represents the

viscous term; and the third term represents large-scale forcing. In our simulations, a constant forcing is applied on the first two shells.

The left boundary conditions (BCs) are $u_{-1} = u_{-2} = 0$. For the fully resolved system, we integrate Equation 1 for shells $n = 0, \dots, N$, where N is chosen with some margin of the dissipative cutoff scale, N_η , as to fully resolve all of the dynamics. In the case of closing the system in an LES fashion, we have to consider a cutoff shell N_c , after which we only have to model the shells $N_c + 1$ and $N_c + 2$ (the left BCs remain the same), as depicted in Figure 1. In all simulations, we set $N_c = 14$, a cutoff chosen within the inertial range but safely above the dissipative cutoff. The velocity fields of the subgrid shells then appear explicitly in the equations for u_{N_c} and u_{N_c-1} :

$$\frac{du_{N_c}}{dt} = i \left(k_{N_c+1} u_{N_c+2} u_{N_c+1}^* - \frac{1}{2} k_{N_c} u_{N_c+1} u_{N_c-1}^* + \frac{1}{2} k_{N_c-1} u_{N_c-1} u_{N_c-2} \right) - \nu k_{N_c}^2 u_{N_c}, \quad (2a)$$

$$\frac{du_{N_c-1}}{dt} = i \left(k_{N_c} u_{N_c+1} u_{N_c}^* - \frac{1}{2} k_{N_c-1} u_{N_c} u_{N_c-2}^* + \frac{1}{2} k_{N_c-2} u_{N_c-2} u_{N_c-3} \right) - \nu k_{N_c-1}^2 u_{N_c-1}. \quad (2b)$$

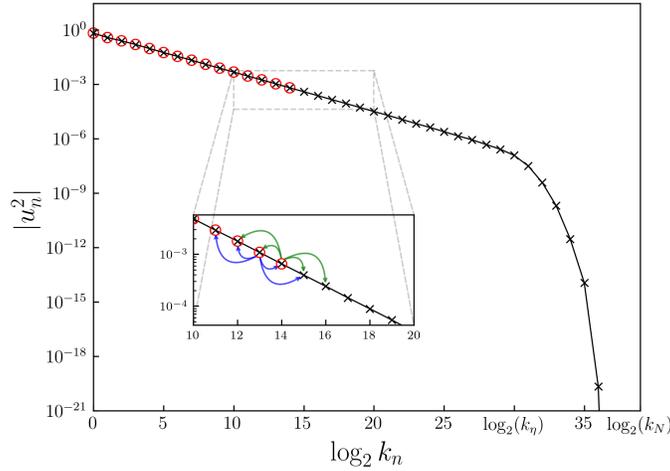


Figure 1: Energy spectrum $|u_n|^2$ as a function of the wavenumber k_n . The red-circled markers indicate the reduced system, while black crosses represent the fully resolved system. The inset zooms into the cutoff region, highlighting the nonlinear energy transfer through triadic interactions, with blue and green arrows indicating interactions for the two shells preceding the cutoff. It becomes clear that to close the system, only the two shells after the cutoff are needed to model. The x-axis is labeled in $\log_2 k_n$, along with the Kolmogorov scale k_η and maximum resolved wavenumber k_N .

3 A posteriori closure

To close the system, we use a deep neural network that predicts the two missing shells, u_{N_c+1} and u_{N_c+2} , using as input the three preceding shells (enough to fix the flux locally). Unlike an *a priori* approach, which trains the model to match instantaneously the subgrid terms, we follow an *a posteriori* or *solver-in-the-loop* strategy, where the neural network operates within the evolving simulation. This allows it to learn not only accurate short-term predictions but also how its own errors propagate over time and influence the system’s long-term behavior.

A key motivation for adopting an *a posteriori* strategy is the presence of distribution shift [31], that is, a mismatch between the statistical distributions encountered during training and deployment. Here, we denote by x the model inputs and by y the corresponding outputs or targets. Three main forms are typically distinguished. (i) *Covariate shift*, where the input distribution changes, $p_{\text{train}}(x) \neq p_{\text{test}}(x)$, while the conditional mapping remains the same, $p(y|x)$; (ii) *Prior or label shift*, where the output distribution differs, $p_{\text{train}}(y) \neq p_{\text{test}}(y)$, but $p(x|y)$ is preserved; and (iii) *Concept shift*, where the underlying relationship between inputs and outputs itself changes, $p_{\text{train}}(y|x) \neq p_{\text{test}}(y|x)$.

In our setting, the shift arises as a dynamically induced *covariate shift*. During solver deployment, small prediction errors in the learned closure accumulate and perturb the system dynamics, thereby altering the distribution of the model inputs—i.e., the resolved variables provided to the closure at each time step. Although the physical relation governing the coupling between resolved and unresolved scales remains invariant, the model gradually encounters regions of phase space that were rare or unseen during training. This phenomenon parallels what is observed in autoregressive generative models, such as language models, where compounding errors cause trajectories to drift away from the data manifold [32].

Embedding the neural network directly within the governing equations during training (the solver-in-the-loop strategy) exposes it to its own prediction-induced state distribution, thus reducing the mismatch between training and deployment distributions. This mitigates the covariate shift, leading to improved stability, generalization, and a more physically consistent closure.

To formalize this approach, let the reduced system $(\tilde{u}_0, \tilde{u}_1, \dots, \tilde{u}_{N_c})$ reside in the *LES manifold* \mathcal{S} , while the full system (u_0, u_1, \dots, u_N) spans the *reference manifold* \mathcal{R} . The governing equations of the shell model define a dynamical system on \mathcal{R} , but large eddy simulations truncate the system to \mathcal{S} , introducing closure errors. To bridge this gap, we introduce a correction operator $\mathcal{C} : \mathcal{S} \rightarrow \mathcal{R}$ parameterized by a neural network, where \mathcal{R} represents the truncated reference manifold, restricted to the resolved scales $(u_0, u_1, \dots, u_{N_c})$. This operator predicts the unresolved shells $\tilde{u}_{N_c+1}, \tilde{u}_{N_c+2} = \mathcal{C}(\tilde{u}_{N_c-2}, \tilde{u}_{N_c-1}, \tilde{u}_{N_c})$, enabling reconstruction of the full state in \mathcal{C} , via an explicit time integration of the missing terms: this corresponds to an additive correction to the resolved terms that are integrated through a fourth-order Runge-Kutta scheme [33].

The operator \mathcal{C} is implemented as a fully connected feedforward neural network (multi-layer perceptron, MLP) with L hidden layers. Given an input vector

$$\mathbf{x} = (\tilde{u}_{N_c-2}, \tilde{u}_{N_c-1}, \tilde{u}_{N_c}) \in \mathbb{C}^3,$$

the output $\tilde{\mathbf{u}} = (\tilde{u}_{N_c+1}, \tilde{u}_{N_c+2}) \in \mathbb{C}^2$ is computed through a sequence of affine transformations and nonlinear activations:

$$\mathbf{h}^{(0)} = \mathbf{x}, \tag{3a}$$

$$\mathbf{h}^{(\ell)} = \sigma(\mathbf{W}^{(\ell)}\mathbf{h}^{(\ell-1)} + \mathbf{b}^{(\ell)}), \quad \text{for } \ell = 1, \dots, L, \tag{3b}$$

$$\tilde{\mathbf{u}} = \mathbf{W}^{(L+1)}\mathbf{h}^{(L)} + \mathbf{b}^{(L+1)}. \tag{3c}$$

Here, $\mathbf{W}^{(\ell)}$ and $\mathbf{b}^{(\ell)}$ are the weight matrix and bias vector for the ℓ -th layer, while $\sigma(\cdot)$ is a nonlinear activation function, in our case ReLU [34]. The final layer has a linear activation to ensure unrestricted output values.

The evolution of the system is governed by a discrete solver \mathcal{P} , which advances the resolved shells via:

$$\tilde{\mathbf{u}}(t + \Delta t) = \mathcal{P}(\tilde{\mathbf{u}}(t), \tilde{u}_{N_c+1}, \tilde{u}_{N_c+2}), \tag{4}$$

where $\tilde{\mathbf{u}}(t) = (\tilde{u}_0, \dots, \tilde{u}_{N_c})|_t$. Unlike *a priori* closure models trained to minimize instantaneous errors, our *a posteriori* approach integrates \mathcal{C} recurrently into \mathcal{P} . During training, the solver interacts with \mathcal{C} over m steps, exposing it to its own error dynamics. The loss function evaluates deviations from the reference trajectory across this horizon:

$$\mathcal{L}(\tilde{\mathbf{u}}^\theta, \mathbf{u}^{\text{ref}}) = \sum_{k=1}^m \|\tilde{\mathbf{u}}^\theta(t + k\Delta t) - \mathbf{u}^{\text{ref}}(t + k\Delta t)\|_2^2, \tag{5}$$

where \mathbf{u}^{ref} is obtained from a direct numerical simulation of \mathcal{R} and $\tilde{\mathbf{u}}^\theta$ is the LES solution with the neural network parameters $\theta = \{\mathbf{W}^{(\ell)}, \mathbf{b}^{(\ell)}\}_{\ell=1}^{L+1}$. The optimization problem is then formulated as:

$$\min_{\theta} \mathcal{L}(\tilde{\mathbf{u}}^\theta, \mathbf{u}^{\text{ref}}).$$

This multi-step optimization ensures \mathcal{C} learns to mitigate error accumulation and distribution shift, as illustrated in Figure 2.

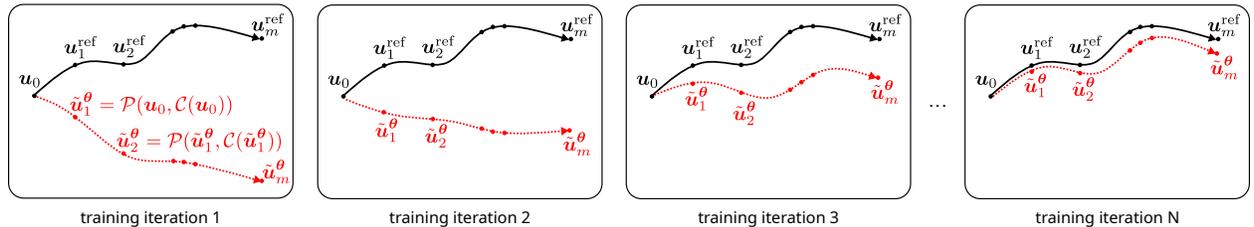


Figure 2: Sketch illustrating the *solver-in-the-loop* approach. The LES trajectory $\tilde{\mathbf{u}}$ is evolved over m time steps using the discrete solver \mathcal{P} , while the correction operator \mathcal{C} , parameterized by a neural network, predicts the unresolved scales. The objective is to iteratively refine \mathcal{C} during training, bringing $\tilde{\mathbf{u}}$ closer to the reference trajectory \mathbf{u}^{ref} , which integrates the full system dynamics from the same initial condition. Here, the subscript denotes the time step index of the trajectory, i.e., $\mathbf{u}_k = \mathbf{u}(t + k\Delta t)$, and not the shell index.

The MLP used for the closure has seven hidden layers of 256 neurons each. Training is performed with the Adam optimiser [35], using a learning rate of 10^{-4} . A detailed summary of the numerical and training parameters as well as the algorithm for training is provided in Appendix A. For more detailed information regarding the training procedure we refer the readers to Ref. [27].

4 Results

In this section, our DL-closed reduced system of equations, hereon referred to as LES-NN, is evaluated by comparing its predictions against the fully resolved ground truth across multiple statistical measures. We analyze its performance in reproducing key turbulence characteristics, including structure functions, temporal evolution of energy, and scale-dependent correlations. Particular attention is given to the ability of the model to capture the tails of joint probability density functions between resolved and unresolved variables; as well as scale invariant statistics following the third hypothesis of K62 theory [29], and whether this symmetry is preserved by the closure. All LES-NN results are obtained starting from initial conditions not used during training, and the trajectories are evolved for time horizons substantially longer than those present in the training data (details in Appendix A). Thus, the form of generalization assessed here is in-distribution: robustness to unseen initial conditions and to long-time evolution, rather than out-of-distribution extrapolation across different Reynolds numbers or cutoff scales.

The structure functions, defined as the difference between velocity increments across different scales to some power p , can be defined for the shell models as

$$S_n^{(p)} = \langle |u_n|^p \rangle, \quad (6)$$

where $\langle \cdot \rangle$ denotes averaging in time and across different initial conditions. These are shown in Figure 3 up to the cutoff shell, for $p = 1, \dots, 6$ for both the fully resolved system (ground truth) and the LES-NN model. There is a great agreement between the model and the ground truth for all orders of structure functions showed, even close to the cutoff scale, which is where the effect of the subgrid scales is more noticeable.

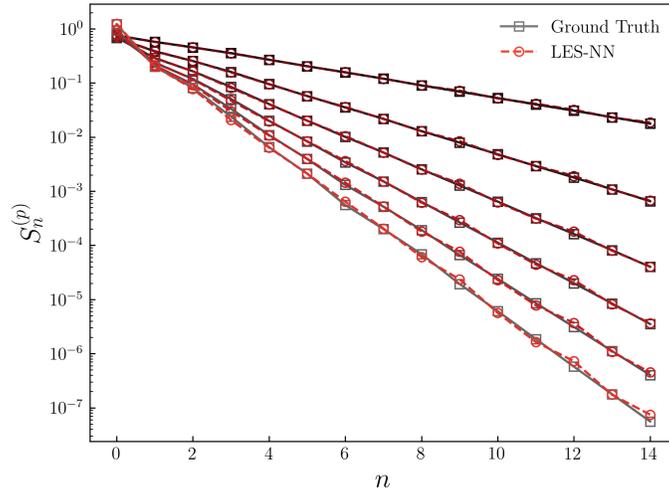


Figure 3: Structure functions $S_n^{(p)}$ as a function of shell index n for different orders $p = 1, \dots, 6$. The LES-NN model is compared with the ground truth solution.

In Figure 4, we show the time evolution of the logarithm of the normalised energy field (velocity field scales as $u_n \sim k_n^{-1/3}$) for both the ground truth (GT) and the LES-NN model, along with the normalized error between the two. Both the GT and LES-NN models start from the same initial condition (IC). It is evident that the LES-NN model rapidly decorrelates from the GT. This behavior is expected, as we cannot synchronize past the Lyapunov time. For this particular IC, the discrepancy is particularly clear because we have chosen an IC that is within a burst of energy that has already propagated to the small scales, where the dynamics are much faster. As a result, the effective Lyapunov time for the LES can be roughly approximated as the time scale of the cutoff shell.

In order to assess the cross-correlations between resolved and unresolved scales, Figure 5 shows the joint PDF between the summed energy of the *resolved* shells used as input to the neural network, $\tilde{u}_{N_c}, \tilde{u}_{N_c-1}, \tilde{u}_{N_c-2}$, and the sum of the

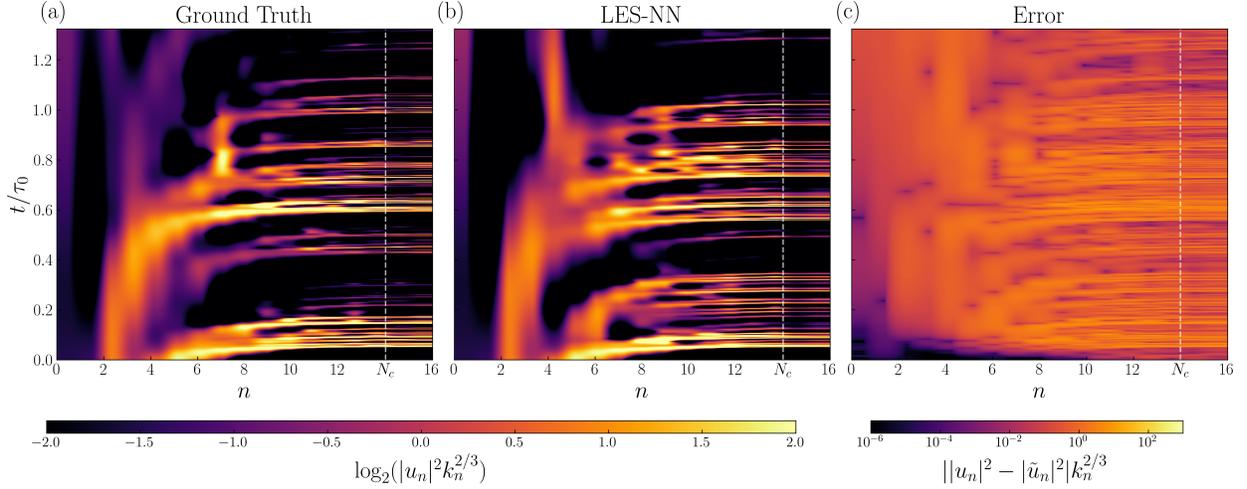


Figure 4: Time evolution of the normalised energy density $\log_2(|u_n|^2 k_n^{2/3})$ for (a) the ground truth and (b) the LES-NN model. (c) Absolute normalised error $||u_n|^2 - |\tilde{u}_n|^2| k_n^{2/3}$ between the two fields. The x-axis represents the wavenumber index n , while the y-axis represents time normalised by the integral time scale t/τ_0 . The dashed line represents the cutoff scale.

unresolved energy, i.e., the sum of the energies of the shells that the NN outputs, $\tilde{u}_{N_c+1}^\theta, \tilde{u}_{N_c+2}^\theta$, for both the ground truth and the LES-NN model. The overall shape of the distribution is well captured, but deviations appear in the joint tails, indicating that the learned closure underestimates the strength of the cross-scale correlations, which are connected with the correlation of the shell phases. We discuss possible causes of this discrepancy at the end of this section.

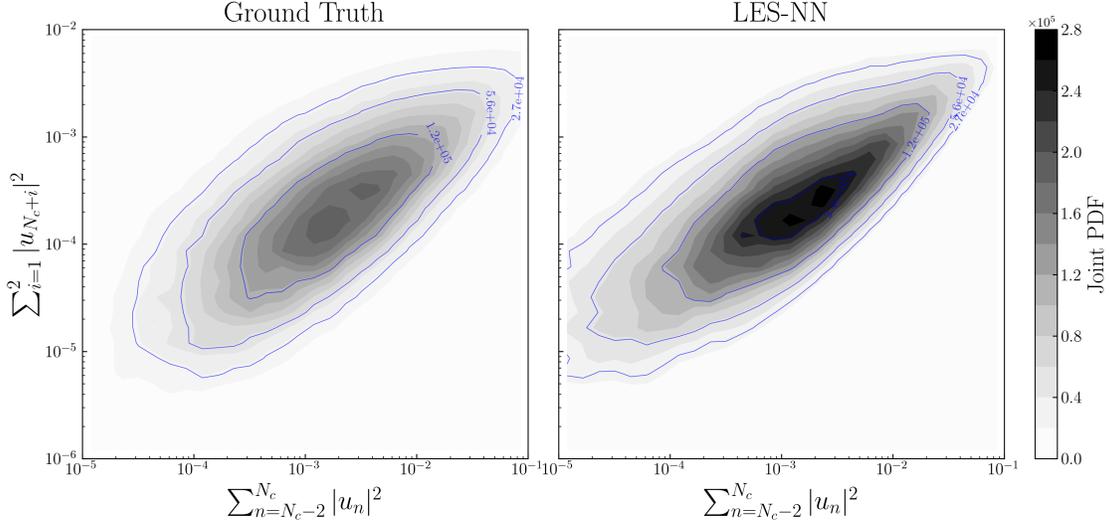


Figure 5: Contours of the joint PDF of the summed energy across the input shells for the neural network (x-axis) and the subgrid-scale output shells (y-axis) for both the ground truth and the LES-NN model.

In his seminal 1962 study [29], Kolmogorov advanced a theoretical framework to characterize the intermittency observed in turbulence at small scales. Central to this framework was the concept of *multipliers*, defined as scale-dependent ratios of velocity increments. These multipliers, denoted as $w_{ij}(x; \ell, \ell')$, were expressed as:

$$w_{ij}(x; \ell, \ell') = \frac{\delta_i v_j(x, \ell)}{\delta_i v_j(x, \ell')}, \quad (7)$$

where $\delta_i v_j(x, \ell) = v_j(x + \ell \mathbf{e}_i) - v_j(x)$ represents the increment of the j -th velocity component across a spatial separation ℓ along the unit vector \mathbf{e}_i . Kolmogorov posited that at sufficiently high Reynolds numbers, the statistical distributions of these multipliers would exhibit universality—dependent solely on the scale ratio ℓ/ℓ' and independent of the absolute scales ℓ or ℓ' . Furthermore, he hypothesized statistical independence between multipliers associated with scales separated by large intervals, a conjecture later termed his third hypothesis [36].

For shell models, one can easily define amplitude and phase multipliers [37], respectively, via:

$$w_n(t) = \left| \frac{u_n(t)}{u_{n-1}(t)} \right|, \quad (8)$$

$$\Delta_n(t) = \varphi_n(t) - \varphi_{n-1}(t) - \varphi_{n-2}(t), \quad (9)$$

where $\varphi_n = \arg u_n$.

In Figure 6, we show the PDF of the amplitude and phase multipliers for the ground truth and our model, LES-NN, for shells $n = 7, \dots, N_c$. For the fully resolved model, we observe the expected collapse. However, for the LES-NN model, the learned closure breaks this symmetry, and the multipliers are no longer scale invariant near the cutoff. This is a limitation of the model.

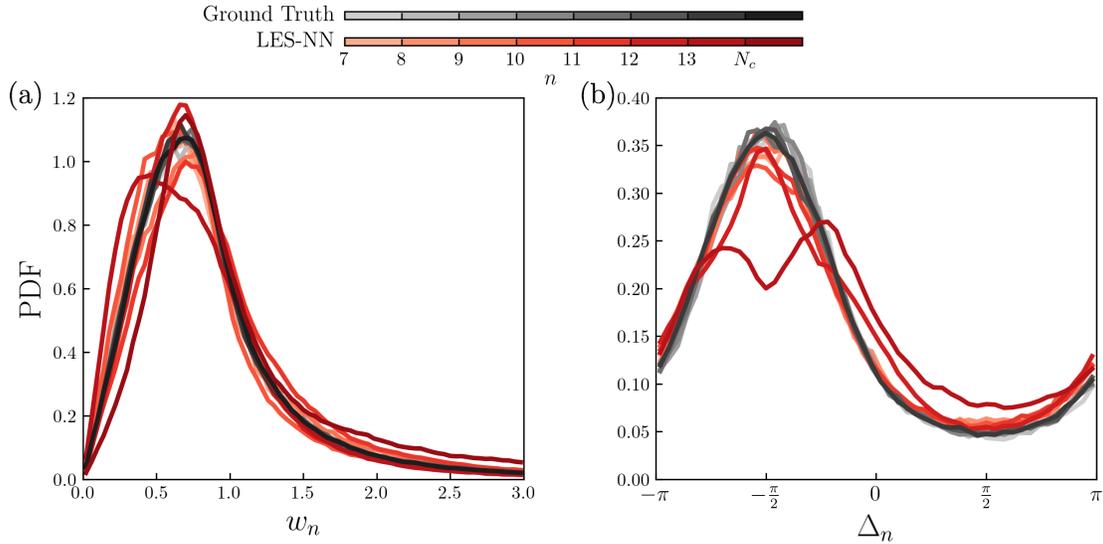


Figure 6: PDF of the (a) amplitude multipliers, w_n , and (b) phase multipliers, Δ_n , for the ground truth and LES-NN model, for the shells 7 to N_c .

We extend this analysis in Figure 7 by examining the joint PDF of the amplitude of the product of the last two resolved multipliers and the product of the two unresolved multipliers (which include the neural network’s predictions). The LES-NN model accurately captures the resolved multipliers but fails significantly in the unresolved ones, leading to incorrect predictions not only in the tails but also in the mean.

To further probe correlations, following Eyink et al. [38], we introduce variables analogous to the spins of the two-dimensional XY model [39, 40],

$$\sigma_n(t) = \ln w_n(t), \quad U_n(t) = \exp(i \Delta_n(t)), \quad (10)$$

where σ_n quantifies local amplitude fluctuations and U_n encodes relative phase differences between neighboring shells. The set $\{U_n\}$ behaves as interacting rotators on the unit circle, whose alignment measures the degree of phase coherence across scales. In this analogy, the energy cascade corresponds to the propagation of order in a spin chain: strong alignment implies coherent phase dynamics and efficient energy transfer. To quantify this behavior, we compute correlation functions,

$$C_{XY}(n, m) = \langle X_n^* Y_m \rangle - \langle X_n^* \rangle \langle Y_m \rangle, \quad (11)$$

for $X = Y = \sigma$ and $X = Y = U$, which measure how amplitude and phase coherence decay with scale separation. The reference shell, $n = 11$, is chosen within the inertial range, close to but below the cutoff. The results are shown in Figure 8 for the GT and LES-NN model. We see the expected fast decay with increasing $|m - n|$; however, the

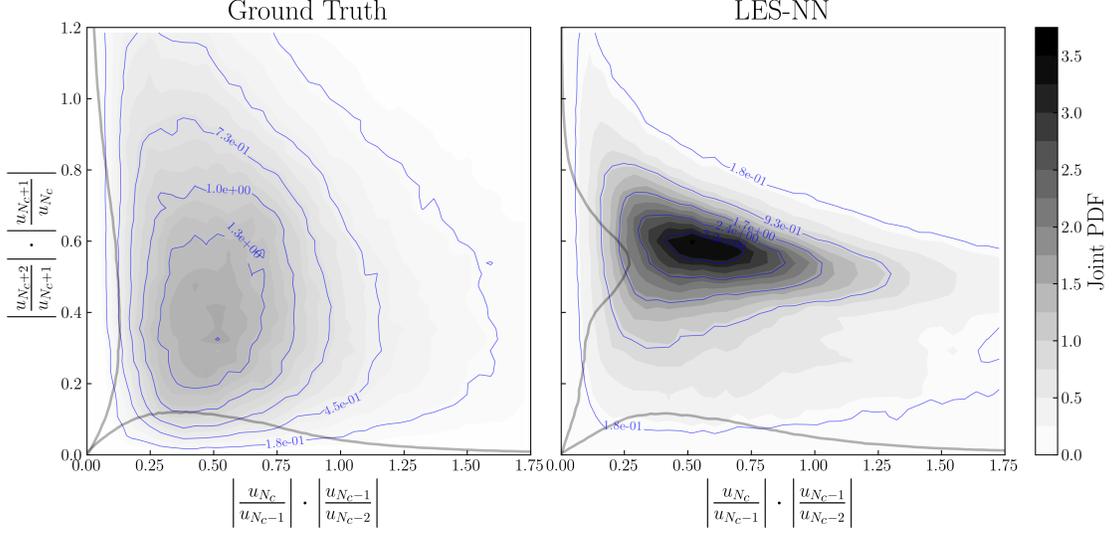


Figure 7: Contours of the joint PDF between the product of the last two resolved multipliers (x-axis) and the product of the two subgrid scale multipliers (y-axis) for both the ground truth and the LES-NN model.

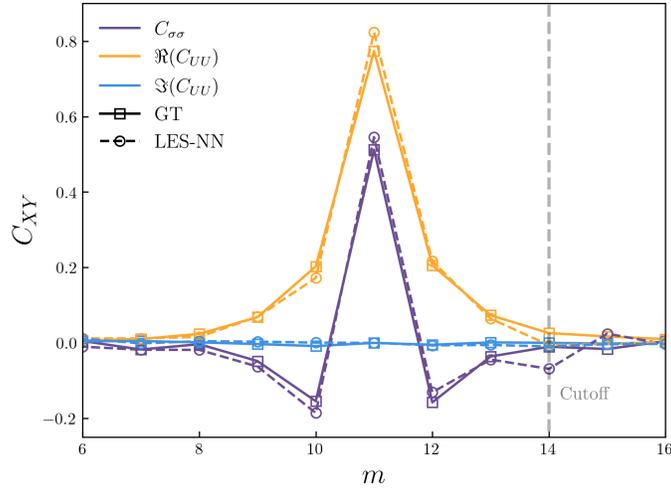


Figure 8: Correlation functions $C_{XY}(n, m)$ for amplitude fluctuations ($X = Y = \sigma$) and phase rotations ($X = Y = U$), using shell $n = 11$ as the reference within the inertial range. The curves show how correlations decay as the shell separation $|m - n|$ increases for both the ground truth and the LES-NN model. The dashed vertical line marks the cutoff shell N_c . While both models display the expected rapid decay at large separations, the LES-NN correlations deviate near the cutoff, indicating a loss of phase and amplitude coherence in this region.

correlations near the cutoff are wrongly estimated for the LES-NN model. This helps shine a light on the lack of collapse of the multipliers near the cutoff.

The results presented above highlight both the strengths and limitations of the learned closure. The LES-NN model reproduces the resolved-scale statistics with high accuracy, including the overall scaling of the structure functions up to high orders, although small deviations appear at the highest orders. However, these residual differences become more evident when we examine observables that explicitly couple resolved and unresolved variables, such as the joint PDFs and the multipliers near the cutoff. In these quantities, the model underestimates the tails of the distributions and fails to recover the full joint variability between scales. This analysis thus acts as a magnifying glass, revealing the origin of the small but systematic departures observed in the higher-order statistics of the resolved variables.

Several factors could in principle contribute to these discrepancies. One possibility is the neural-network architecture itself: a fully connected MLP may not possess sufficient expressive power to capture all higher-order correlations, and spectral bias could suppress the small-scale variability associated with strong intermittency. However, in our tests, changing the network architecture and increasing its depth or width (above what we considered for the results presented here) did not lead to significant improvements. Similarly, changing the loss function to include the subgrid shells explicitly, rather than only the resolved variables, did not alter the results. These observations suggest that the origin of the discrepancies does not lie primarily in architectural choices or in the omission of subgrid shells from the loss function.

Instead, our findings point to a structural limitation arising from the formulation of the closure problem itself, particularly the Markovian nature of the model. The Mori–Zwanzig (MZ) formalism [41, 42] provides a natural perspective on this issue. When a high-dimensional dynamical system is projected onto a reduced set of resolved variables (as it is done in LES), the resulting exact evolution equation contains two additional contributions beyond the Markovian term: a memory kernel, encoding the delayed interactions between resolved and unresolved variables, and an orthogonal dynamics term that accounts for unresolved-unresolved interactions that cannot be reproduced with the memory term and are usually modeled as noise. A purely Markovian closure omits both effects, and thus misses key ingredients that should in principle help in capturing the correlations between resolved and unresolved variables. This structural lack of memory offers a possible explanation for the breakdown of scale-invariant multiplier statistics near the cutoff and the incomplete reconstruction of resolved-unresolved correlations observed in our results.

Possible ways forward include extending the framework to incorporate non-Markovian effects in the spirit of the MZ formulation, for instance by learning effective memory-kernel or orthogonal-dynamics terms directly from data, as recently explored in data-driven MZ approaches [43, 44]. In parallel, one may also improve the closure by explicitly incorporating the known scale invariance of multipliers into the learning process, either as a penalty in the loss function or as hard constraint on the model architecture, so that the collapse of the multiplier PDFs is enforced rather than left as an emergent property. Another alternative is to reformulate the problem in terms of multipliers themselves by rescaling the governing equations, as done by Domingues Lemos et al. [24, 25] within the hidden-symmetry framework [26]. Combining these ideas—for example, by training a solver-in-the-loop model directly in the multiplier variables or by enforcing multiplier symmetries within the present formulation—may lead to closures that simultaneously reproduce high-order resolved statistics and respect the scale-invariant structure of the multipliers. These considerations are an attempt at clarifying the origin of the observed discrepancies and point to concrete directions for improving the closure framework.

5 Conclusion

In this work, we have examined in detail a data-driven closure for a shell model of turbulence trained with a solver-in-the-loop (a posteriori) strategy [27]. Building on our previous study, which focused mainly on high-order statistics of the resolved variables, we have gone beyond single-scale observables and analysed quantities that explicitly couple resolved and unresolved degrees of freedom. In particular, we investigated joint probability density functions between resolved and subgrid shells, Kolmogorov multipliers and their scale invariance, and correlation functions inspired by the XY-model analogy [38–40].

Our results show that our data-driven closure reproduces resolved-scale statistics with high accuracy, including structure functions up to order $p = 6$, and remains stable under long-time integration. However, when we probe different observables that evaluate the correlations between resolved and unresolved shells, systematic discrepancies emerge. The joint PDFs between resolved and unresolved energies reveal an underestimation of the tails, indicating that the model does not fully capture the variability across the cutoff. Similarly, the amplitude and phase multipliers lose their expected scale invariance near the cutoff, and the spin–spin correlations involving shells close to the cutoff are slightly misestimated. These diagnostics act as a magnifying glass: they expose limitations that are not apparent from the resolved-scale statistics alone.

We have argued that these shortcomings are unlikely to be resolved merely by increasing network capacity or by including subgrid shells directly in the loss function, and instead point to a possible structural limitation of a Markovian closure. Within the Mori–Zwanzig formalism [41, 42], an exact coarse-grained description necessarily involves a memory kernel and an orthogonal dynamics term. A memoryless closure, such as the one considered here, cannot fully reproduce these effects, which offers a plausible explanation for the breakdown of multiplier scale invariance near the cutoff and the incomplete reconstruction of resolved-unresolved correlations.

These findings suggest several directions for future work. On the one hand, incorporating non-Markovian effects in a principled way, for example via data-driven MZ operators [43, 44], may allow closures that better capture multi-scale correlations across the cutoff. On the other hand, explicitly embedding key symmetries of turbulence into the

learning problem—such as the scale invariance of multipliers, or working directly in the multiplier/hidden-symmetry formulation [24–26]—could lead to models that are both more universal and less sensitive to the choice of cutoff. Extending and testing these ideas in three-dimensional Navier–Stokes turbulence, where similar rescaling procedures can be defined [45], remains an important challenge. Overall, our analysis suggests that combining solver-in-the-loop training with a physically informed structure is a promising route towards robust and better-performing data-driven subgrid scale closures.

Acknowledgments

The authors benefited from discussions with G. Eyink, A. Mailybaev, M. Sbragaglia. This research was supported by European Union’s HORIZON MSCA Doctoral Networks programme under Grant Agreement No. 101072344, project AQTIVATE (Advanced computing, Quantum algorithms and data-driven Approaches for science, Technology and Engineering), the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme Smart-TURB (Grant Agreement No. 882340), and through an Inria Chair.

References

- [1] Uriel Frisch. *Turbulence: The legacy of A. N. Kolmogorov*. Cambridge University Press, 1995.
- [2] Peter A. Davidson, Yukio Kaneda, Keith Moffatt, and Katepalli R. Sreenivasan, editors. *A Voyage Through Turbulence*. Cambridge University Press, 2011.
- [3] Katepalli R. Sreenivasan and Jörg Schumacher. What is the turbulence problem, and when may we regard it as solved? *Annual Review of Condensed Matter Physics*, 16(Volume 16, 2025):121–143, 2025.
- [4] Gregory Falkovich and Katepalli R. Sreenivasan. Lessons from hydrodynamic turbulence. *Physics Today*, 59(4):43–49, 2006.
- [5] Parviz Moin and Krishnan Mahesh. Direct Numerical Simulation: a tool in turbulence research. *Annual Review of Fluid Mechanics*, 30(Volume 30, 1998):539–578, 1998.
- [6] John Kim and Anthony Leonard. The early days and rise of turbulence simulation. *Annual Review of Fluid Mechanics*, 56(Volume 56, 2024):21–44, 2024.
- [7] Charles Meneveau and Joseph Katz. Scale-invariance and turbulence models for large-eddy simulation. *Annual Review Fluid Mechanics*, 2000.
- [8] J. Smagorinsky. General circulation experiments with the primitive equations: I. the basic experiment. *Monthly Weather Review*, 91(3):99 – 164, 1963.
- [9] D. K. Lilly. The representation of small-scale turbulence in numerical simulation experiments. *Proceedings of the IBM Scientific Computing Symposium on Environmental Sciences*, pages 195–210, 1967.
- [10] R. H. Kraichnan. Eddy viscosity in two and three dimensions. *Journal of the Atmospheric Sciences*, 33:1521–1536, August 1976.
- [11] M Lesieur and O Metais. New trends in large-eddy simulations of turbulence. *Annual Review of Fluid Mechanics*, 28(Volume 28, 1996):45–82, 1996.
- [12] Luca Biferale, Alexei A. Mailybaev, and Giorgio Parisi. Optimal subgrid scheme for shell models of turbulence. *Physical Review E*, 95, 2017.
- [13] Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, 51(Volume 51, 2019):357–377, 2019.
- [14] Michele Buzzicotti. Data reconstruction for complex flows using AI: Recent progress, obstacles, and perspectives. *Europhysics Letters*, 142(2):23001, apr 2023.
- [15] A. Crisanti, M. H. Jensen, A. Vulpiani, and G. Paladin. Intermittency and predictability in turbulence. *Phys. Rev. Lett.*, 70:166–169, Jan 1993.
- [16] Giorgio Parisi and Uriel Frisch. On the singularity structure of fully developed turbulence. In M. Ghil, R. Benzi, and G. Parisi, editors, *Turbulence and Predictability in Geophysical Fluid Dynamics and Climate Dynamics*, pages 84–87. North-Holland, Amsterdam, 1985.
- [17] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [18] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.
- [19] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.
- [20] Zachary C. Lipton. The mythos of model interpretability. *Communications of the ACM*, 61(10):36–43, 2018.
- [21] Yoshua Bengio and Yann LeCun. Scaling learning algorithms towards AI. *Large-Scale Kernel Machines*, pages 321–360, 2007.
- [22] Luca Biferale. Shell models of energy cascade in turbulence. *Annual Review of Fluid Mechanics*, 35(1):441–468, 2003.
- [23] Giulio Ortali, Alessandro Corbetta, Gianluigi Rozza, and Federico Toschi. Numerical proof of shell model turbulence closure. *Physical Review Fluids*, 7, 2022.
- [24] J. Domingues Lemos and A. A. Mailybaev. Data-based approach for time-correlated closures of turbulence models. *Phys. Rev. E*, 109:025101, Feb 2024.
- [25] Julia Domingues Lemos and Fabio Pereira dos Santos. Statistical machine learning tools for probabilistic closures of turbulence models, 2025.

- [26] Alexei A. Mailybaev. Hidden scale invariance of intermittent turbulence in a shell model. *Phys. Rev. Fluids*, 6:L012601, Jan 2021.
- [27] André Freitas, Kiwon Um, Mathieu Desbrun, Michele Buzzicotti, and Luca Biferale. Solver-in-the-loop approach to closure of shell models of turbulence. *Phys. Rev. Fluids*, 10:044602, Apr 2025.
- [28] Kiwon Um, Robert Brand, Yun Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative PDE-solvers. *Advances in Neural Information Processing Systems*, 2020.
- [29] A. N. Kolmogorov. A refinement of previous hypotheses concerning the local structure of turbulence in a viscous incompressible fluid at high Reynolds number. *Journal of Fluid Mechanics*, 13(1):82–85, 1962.
- [30] Victor S. L’vov, Evgenii Podivilov, Anna Pomyalov, Itamar Procaccia, and Damien Vandembroucq. Improved shell model of turbulence. *Physical Review E*, 58:1811–1822, 1998.
- [31] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence, editors. *Dataset Shift in Machine Learning*. The MIT Press, Cambridge, MA, 2009.
- [32] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020.
- [33] Martin Wilhelm Kutta. Beitrag zur näherungsweise integration totaler differentialgleichungen. *Zeitschrift für Mathematik und Physik*, 46:435–453, 1901.
- [34] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML’10, page 807–814, Madison, WI, USA, 2010. Omnipress.
- [35] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.
- [36] Qiaoning Chen, Shiyi Chen, Gregory L. Eyink, and Katepalli R. Sreenivasan. Kolmogorov’s third hypothesis and turbulent sign statistics. *Phys. Rev. Lett.*, 90:254501, Jun 2003.
- [37] R. Benzi, L. Biferale, and G. Parisi. On intermittency in a cascade model for turbulence. *Physica D: Nonlinear Phenomena*, 65(1):163–171, 1993.
- [38] G. L. Eyink, S. Chen, and Q. Chen. Gibbsian hypothesis in turbulence. *Journal of Statistical Physics*, 2003.
- [39] V. L. Berezinskii. Destruction of Long-range Order in One-dimensional and Two-dimensional Systems having a Continuous Symmetry Group I. Classical Systems. *Sov. Phys. JETP*, 32:493–500, 1971.
- [40] J M Kosterlitz and D J Thouless. Ordering, metastability and phase transitions in two-dimensional systems. *Journal of Physics C: Solid State Physics*, 6(7):1181, apr 1973.
- [41] Hazime Mori. Transport, collective motion, and Brownian motion. *Progress of Theoretical Physics*, 33(3):423–455, 03 1965.
- [42] Robert Zwanzig. Ensemble method in the theory of irreversibility. *The Journal of Chemical Physics*, 33(5):1338–1341, 11 1960.
- [43] Yen Ting Lin, Yifeng Tian, Danny Perez, and Daniel Livescu. Regression-based projection for learning Mori–Zwanzig operators. *SIAM Journal on Applied Dynamical Systems*, 22(4):2890–2926, 2023.
- [44] Xander de Wit, Alessandro Gabbana, Michael Woodward, Yen Ting Lin, Federico Toschi, and Daniel Livescu. Is memory all you need? Data-driven Mori-Zwanzig modeling of Lagrangian particle dynamics in turbulent flows, 2025.
- [45] Alexei A. Mailybaev and Simon Thalabard. Hidden scale invariance in Navier–Stokes intermittency. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 380(2218):20210098, 2022.

A Training details

The numerical simulations and solver-in-the-loop training procedure were performed using the parameters listed in Table 1. The corresponding training loop is summarized in Algorithm 1. The algorithm represents a single training iteration of the solver-in-the-loop scheme. At each time step, the neural network predicts the unresolved shells, $\tilde{u}_t^{>\theta}$, from the resolved state $\tilde{u}_t^{<}$. The resolved dynamics are then advanced in time, and the closure terms \mathcal{C}_{N_c-1} and \mathcal{C}_{N_c} , which correspond to the contributions excluded from the direct integration because they depend on the predicted unresolved shells, are subsequently added through explicit integration. This procedure is repeated for `msteps` iterations, producing trajectories used to evaluate the multi-step loss that measures the deviation from the reference solution. Gradients of the loss with respect to the neural-network parameters are then obtained via backpropagation, and the parameters are updated using gradient-based optimization. All code developed for this work is openly available at github.com/andremfreitas/sm_loop.

Table 1: Values of the parameters used in the numerical experiments.

Parameter	Value	Description
ν	1×10^{-12}	Viscosity
Re	$\approx 10^{12}$	Reynolds number
ϵ	0.5	Forcing amplitude
N	40	Number of shells
N_η	30	Kolmogorov scale
N_c	14	Subgrid cutoff scale
τ_0	7.553×10^{-1}	Eddy turnover time at the integral scale
τ_η	1.8367×10^{-6}	Eddy turnover time at the dissipative scale
Δt	1×10^{-8}	Timestep of ground truth simulation
$\Delta \tilde{t}$	1×10^{-5}	Timestep of LES-NN model
N_{data}	256	Number of initial conditions in the dataset
N_{batch}	1024	Batch size used for training
T_{train}	$1.65 \tau_0$	Integration time of training dataset
T_{test}	$3.31 \tau_0$	Integration time of test dataset
<code>msteps</code>	100	Time steps in-the-loop per training iteration

Algorithm 1 Training loop (single training iteration).

1: Initialize gradient tape	▷ Record operations for differentiation
2: $\tilde{u} \leftarrow \tilde{u}_0$	▷ Batch of initial conditions sampled from dataset
3: for $t = 0$ to $msteps - 1$ do	
4: $\tilde{u}_t^{>\theta} \leftarrow \text{NN}_\theta(\tilde{u}_t^{<})$	▷ Predict unresolved shells via neural network
5: $\tilde{u}_{t+1}^{<} \leftarrow \text{RK4}(\tilde{u}_t^{<})$	▷ Advance resolved shells with fourth-order Runge-Kutta
6: $\mathcal{C}_{N_c-1} \leftarrow \Delta \tilde{t} i(ak_{N_c} \tilde{u}_{N_c+1}^\theta \tilde{u}_{N_c}^*)$	▷ Compute closure term for shell $N_c - 1$
7: $\mathcal{C}_{N_c} \leftarrow \Delta \tilde{t} i(ak_{N_c+1} \tilde{u}_{N_c+2}^\theta \tilde{u}_{N_c+1}^{*,\theta} + bk_{N_c} \tilde{u}_{N_c+1}^\theta \tilde{u}_{N_c-1}^*)$	▷ Compute closure term for shell N_c
8: $\mathcal{C} \leftarrow \text{concatenate}(\mathcal{C}_{N_c-1}, \mathcal{C}_{N_c})$	▷ Aggregate closure corrections
9: $\tilde{u}_{t+1}^{<} \leftarrow \tilde{u}_{t+1}^{<} + \mathcal{C}$	▷ Update resolved shells with closure contribution
10: end for	
11: Compute loss \mathcal{L}	▷ Compare predicted and reference trajectories
12: Compute gradients $\nabla_\theta \mathcal{L}$	▷ Differentiate loss w.r.t. network parameters
13: Update parameters θ	▷ Apply optimizer step