GENERATIVE SUBGRID-SCALE MODELING

Jiaxi Zhao
National University of Singapore
jiaxi.zhao@u.nus.edu

Sohei Arisaka
National University of Singapore & Kajima Corporation
sohei.arisaka@u.nus.edu

Qianxiao Li
National University of Singapore
qianxiao@nus.edu.sg

ABSTRACT

The mismatch between the a-priori and a-posteriori error is ubiquitous in data-driven subgrid-scale (SGS) modeling, which is an important ingredient in large eddy simulations. In this work, we investigate the cause of this mismatch in depth and attribute it to two issues: data imbalance and multi-valuedness. Based on this understanding, we propose a generative modeling approach for the SGS stresses that resolves the issue of multi-valuedness and demonstrate its effectiveness in the Kuramoto-Sivashinsky equation.

1 INTRODUCTION

Modern computational fluid dynamics applications demand the simulation of fluid equations on meshes ranging from millions to billions of elements. For most real-world simulation problems such as wind farms and turbomachines, direct numerical simulations that resolve the fluid physics at all scales are still impractical on current supercomputers. Large eddy simulations reduce computational cost by separating fluid physics into two scales. While the physics of scales larger than the mesh resolution are resolved by discretizing and simulating the equation, those smaller scale physics below the mesh resolution have to be captured by subgrid-scale (SGS) modeling, which employs a phenomenological or data-driven model to resolve the errors introduced by the coarse scales.

More concretely, we study the SGS modeling of two equations: the incompressible Navier-Stokes (NS) equations

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j}, \quad \frac{\partial u_i}{\partial x_i} = 0, \tag{1}$$

and the Kuramoto-Sivashinsky (KS) equation

$$\begin{aligned} u_t &= -(c+u)u_x - uu_x - u_{xx} - \nu u_{xxxx}, \\ u(0,t) &= u(L,t) = u_x(0,t) = u_x(L,t) = 0. \end{aligned} \tag{2}$$

The NS equations represent the practical applications of wide interest, while the KS equation serves as a toy model to test our insights and algorithms. The KS equation is chosen as a test as it also exhibit chaotic behavior much like fluid turbulence, but it is lower dimensional and thus easier to visualize and simulate.

Consider a spatially-homogeneous filter G, e.g. Gaussian filter $G(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^3}} e^{-\|\mathbf{x}\|_2^2/2}$, convolved to the field variable $\mathbf{u}$ componentwise, i.e. $\widetilde{\mathbf{u}} = G * \mathbf{u}$, one can obtain the equation governing the filtered field variable $\widetilde{\mathbf{u}}$:

$$\frac{\partial \widetilde{u}_i}{\partial t} + \widetilde{u}_j \frac{\partial \widetilde{u}_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \widetilde{p}}{\partial x_i} + \nu \frac{\partial^2 \widetilde{u}_i}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j}, \quad \frac{\partial \widetilde{u}_i}{\partial x_i} = 0. \tag{3}$$

Most of the linear operations commute with spatially-homogeneous convolution while the nonlinear convection term $u_j \frac{\partial u_i}{\partial x_j}$ does not, resulting in an unclosed term τ, the so-called SGS stress given explicitly by

$$\tau_{ij} = \widetilde{u_i u_j} - \widetilde{u}_i \widetilde{u}_j. \tag{4}$$

The goal of the SGS modeling is to build a closure model to calculate the SGS stress from the filtered velocity field. For example, the well-known Smagorinsky model provides $\tau = -2C_s\Delta^2|\widetilde{S}|\widetilde{S}$, where $\widetilde{S} = \frac{1}{2}(\nabla\widetilde{\mathbf{u}} + (\nabla\widetilde{\mathbf{u}})^T)$ is the rate of strain tensor, Δ is the grid scale, and C_s is the Smagorinsky constant. While empirical methods can be successful when appropriate SGS model classes and their parameters are selected, making such choices in practical applications is still challenging and requires not only domain knowledge but also trial-and-error.

In recent years, researchers have been exploring using neural network to "learn" the SGS model from the high fidelity data (Maulik & San, 2017; Park & Choi, 2021). Among most of these works, the data-driven surrogate turbulence model is determined via minimizing a so-called a-priori error defined as the deviation between the estimated SGS stresses calculated from the model and those calculated from the high fidelity simulations, although parametrization and regularization can be very different. However, a common observation in Sanderse et al. (2024) is that the a-posteriori error does not have similar trend as that of the a priori error. Namely, for a data-driven model achieving smaller a-priori error than the classical Smagorinsky model, the a-posteriori error can be much worse than the simulation deploying Smaogorinsky model. This is named as "a-priori and a-posteriori dichotomy" in Maulik et al. (2019) and several works are attempting to resolve this, e.g. Zhao & Li (2024) introduced regularization based on the dynamical information to mitigate the mismatch.

The contribution of this work is two-fold:

1. We explain the mismatch of "a-priori and a-posteriori dichotomy", by studying the dataset for SGS model training and identify two features: data imbalance and multi-valuedness.
2. We propose a generative modeling of the SGS stresses via a conditional Gaussian model to resolve the multi-valuedness and demonstrate improvement in the simulation of the KS equation.

2 A-PRIORI ERROR ANALYSIS FOR REGRESSION-BASED DATA-DRIVEN SGS MODELING

In this section, we briefly discuss regression-based SGS modeling and highlight two key phenomena observed in training the data-driven SGS models: data imbalance and multi-valuedness. The simulation data of the NS equations is generated by OpenFOAM (Weller et al., 1998) and of the KS equation is generated by custom implementation, with details in appendix A.1.

Learning the data-driven SGS model can be formalized as the optimization of the following regression problem:

$$\min_{\theta} \sum_{n} \left\| \phi_\theta(\widetilde{\mathbf{u}}^{(n)}) - \tau^{(n)} \right\|^2, \tag{5}$$

given the parametrized model ϕ_θ and the dataset $\{\widetilde{\mathbf{u}}^{(n)}, \tau^{(n)}\}$ where n denotes the sample. One can specify different choices of input features, and here we fix this to $\widetilde{\mathbf{u}}^{(n)}$ for simplification.

However, it has been pointed out in several works (Wang et al., 2024; Sanderse et al., 2024) on the data-driven SGS modeling that the closure problem is not a well-defined regression problem. Even the same filtered velocity can correspond to different SGS stresses, because it may arise from different underlying flow fields. One evidence is that both the training and test error of the regression problem eq. (5) cannot be optimized to a low threshold, even if an overparametrized model is used. In our experiments, we also observe a substantial training error in optimizing the neural network surrogate SGS model. We perform an ablation study to find that even using model with large capacity (e.g. deep neural networks and XGBoost (Chen & Guestrin, 2016)) cannot reduce the training error to a satisfactory level (see table 1). To explain this observation, in the next two subsections we discuss two key features of the SGS modeling dataset.

2.1 THE ISSUE OF DATASET IMBALANCE

The first feature is that the data distribution is imbalanced. Namely, most of the data is concentrated in the region where both the input and output channels are very small. To illustrate this point, we plot

Table 1: A-priori analysis using channel flow dataset for wall turbulence. The row denotes the input features where $\widetilde{\mathbf{u}}, \nabla\widetilde{\mathbf{u}}, \widetilde{S}, y^+$ are the filtered velocity, shear stresses, rate of strain, and wall unit. The outputs are the SGS stresses of dimension 9. We report the relative MSE on test dataset with models of different number of parameters. This error is similar to that of the training dataset. The dataset contains 1M data pairs. As can be concluded from the table, even if we use XGBoost with parameters number more than the size of the dataset, it cannot over-fit the dataset to achieve a near-zero training loss. We use fully-connected neural networks with sizes of hidden layers specified in the table and for XGBoost we specify the [n_estimators, max_depth].

	# params	$\widetilde{\mathbf{u}}$	$\nabla\widetilde{\mathbf{u}}$	$\widetilde{S}$	$\widetilde{\mathbf{u}}, y^+$	$\nabla\widetilde{\mathbf{u}}, y^+$	$\widetilde{S}, y^+$
NN, $[64]$	877	0.896	0.616	0.772	0.824	0.598	0.729
NN, $[64, 64]$	5037	0.884	0.601	0.775	0.825	0.568	0.715
NN, $[64, 64, 64]$	9708	0.923	0.586	0.774	0.831	0.570	0.690
XGBoost, $[100, 5]$	55910	0.888	0.618	0.759	0.803	0.597	0.721
XGBoost, $[1000, 5]$	558610	0.887	0.539	0.726	0.793	0.515	0.674
XGBoost, $[1000, 10]$	11055275	0.888	0.472	0.712	0.789	0.432	0.728

the histogram of one of the components of the shear stresses $\nabla\widetilde{\mathbf{u}}$ and stress tensor τ of NS equations and $\partial_{xx}u, \tau$ of KS equation in fig. 1. While it is the natural to have both inputs and outputs clustered around 0, but the concentration is so severe as can be read from the histogram plot of the density. Therefore, the data having nontrivial behavior can not be captured in the regression problem as they only occupies a very small portion of the dataset.

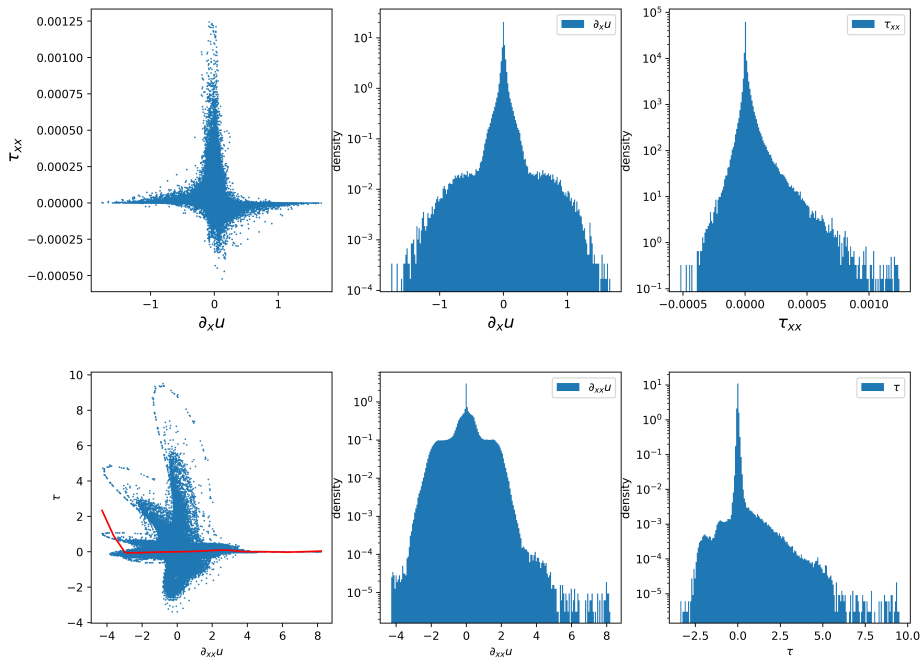


Figure 1: We show both the scattering plot and the histogram of the SGS dataset for NS and KS equations. The upper (lower) row is the dataset for NS (KS) equations respectively. For NS equations, we only show the histogram of the first components of the shear $\nabla\mathbf{u}$ and SGS stresses τ, e.g. $\partial_x u, \tau_{xx}$. Notice that the y (density) axis is in log scale and there exists a peak around 0, indicating that a large potion of data has negligible magnitude. This phenomenon can also be understood as a multiscale structure of the dataset where most of the data has rather smaller scale compared to those large-scale rare data. The issue of multi-valuedness can be directly observed from the leftmost figure of the KS dataset: the scattering pattern between the input and output is quite complicated.

2.2 THE ISSUE OF MULTI-VALUEDNESS

Besides the data imbalance, we observe that the dataset is multivalue, which is more related to the cause of the large training error. The scattering plot of the data pair in both equations are shown in fig. 1, e.g. two figures on the left.

In the lower left figure which corresponds to the 1D KS equation, both the input and the output are 1D and there does not exists a clear dependency between them. Moreover, from this figure we find there exists a set of stresses value corresponding to the same input, implying the issue of multi-valuedness. Under this situation, minimizing the L^2 error between the true stresses and the data-driven model will not result in a training error close to zero as the minimizer is achieved by the average of stresses corresponding to the same inputs. This point is illustrated by the red curve, which corresponds to the learned regression relation of a data-driven model. We observe that the curve only captures the bulk behavior of the dataset concentrating around zero while some large fluctuations of the dataset are ignored. For NS equation, we show the first components of shear stresses and SGS stresses, e.g. $\partial_x u, \tau_{xx}$ in the upper left figure of fig. 1 which also has some complex patterns beyond deterministic relation. Notice that the SGS modeling of NS equations has both input and output of multiple dimension and we have to perform some dimension reduction and nearest neighbor technique to explore the multi-valuedness. The details are explained in appendix A.2. Nevertheless, we also observe the multivalue phenomena in the dataset of the SGS modeling for NS equations.

While the issue of data imbalance revealing certain multiscale structures in the SGS modeling dataset may cause the abnormality and failure in the training of data-driven SGS models, the issue of multi-valuedness implies the intrinsic drawback of the regression-based SGS modeling and we will spend the next section discuss our solution to this issue by the generative modeling.

3 GENERATIVE MODELING FOR THE SGS MODELING

In this section, we propose our method to resolve the mismatch between the a-priori and a-posteriori error of the data-driven SGS modeling by generative modeling. In the previous section, we visualize the dataset of SGS modeling and demonstrated a significant issue of multi-valuedness, causing the large training error. To capture the multi-value nature of the SGS models, a direct solution is to model the SGS stresses as a probability distribution conditioned on the input features, instead of a single value in regression, i.e.

$$\tau = \phi_\theta(u) \quad \rightarrow \quad \tau \sim p_\theta(\cdot|u). \tag{6}$$

As the KS equation is 1D, we do not use the boldface of u, τ for simplicity. Compared with the regression-based model where the value of the stresses is fix by the regression function, generative SGS model provides an ensemble of stresses value obeying certain probability law, thus is consistent with the observation that the dataset is multivalued. Moreover, the correlation coefficients of inputs and outputs are calculated to be very close to zero. While deterministic regression relations may be reasonable for input-output pairs with strong correlations (absolute value of correlation coefficients close to 1), probabilistic relations are better to model the case where the correlation is very weak.

Next, we use the KS equation as a toy model to present our algorithm concretely. The SGS modeling task in the KS equation scenario is to learn a SGS model and solve the KS equation on a much coarser grid with the correction given by the SGS model so that it can simulate this chaotic dynamics correctly in the statistical sense. To evaluate the a-posteriori error of our SGS models, we consider the first two statistics of the flow over space and time as well as the pointwise accuracy:

$$\begin{aligned} \overline{u} = \frac{1}{LT}\int_0^L \int_0^T u(x,t)\, dt\, dx, \quad \overline{u^2} = \frac{1}{LT}\int_0^L \int_0^T u(x,t)^2\, dt\, dx, \\ \|u - u_0\|_2^2 = \frac{1}{LT}\int_0^L \int_0^T (u(x,t) - u_0(x,t))^2\, dt\, dx. \end{aligned} \tag{7}$$

The u_0 denotes the ground truth simulation obtained from fine-grid simulation without any SGS model. The pointwise error is used to demonstrate that it does not make sense to require the trajectorywise accuracy of the SGS modeled simulation and we focus more on the first two moment as in Blonigan & Wang (2014).

3.1 CONDITIONAL GAUSSIAN MODEL

We consider using the conditional Gaussian distribution for the SGS stresses, modeling the stresses as a Gaussian distribution depends on the input features. Instead of minimizing the mean squared error as in eq. (5), we use maximum likelihood principle to train the model, i.e. given a conditional model $p_\theta(\tau|u)$ we maximize the following function

$$\max_\theta \sum_{i=n}^{N} \log p_\theta(\tau^{(n)}|u^{(n)}). \tag{8}$$

Under the assumption that the parametrized family is Gaussian with learnable mean and variance, the loss function is simplified to:

$$\min_\theta \sum_{n=1}^{N} \frac{(\tau^{(n)} - \mu_\theta(u^{(n)}))^2}{2(\sigma_\theta(u^{(n)}))^2} + \log \sigma_\theta(u^{(n)}) \tag{9}$$

Specifically, the output dimension is given by 2 with one dimension represents the mean μ while the other dimension represents the standard deviation σ which is guaranteed to be positive by applying a softplus activation.

Compared to the regression-based approach which can be understood as approximating the mean value, the conditional Gaussian distributions also characterize the variance at each point that implies the severeness of the multi-valuedness at different inputs. Different from the regression-based SGS model, our generative SGS model is trained via maximum likelihood. Its output is a probabilistic distribution, or a sample from this distribution. Consequently, there exists several choices to plug this generative SGS models into the simulator and quantify its a-posteriori error, among which we emphasize two notions: temporal and spatial consistency.

3.2 TEMPORAL AND SPATIAL CONSISTENCY

As we are using a local SGS model for transient simulation, at each spatial-temporal point (x, t) this model is used to predict the stresses

$$\tau(x, t) \sim p_\theta(\cdot|u(x, t)). \tag{10}$$

Therefore, when sampling from these distributions to perform simulation, two kinds of consistency arises. Spatial (temporal) consistency refers to the consistency of generative models at different spatial (temporal) points respectively. For example, to enforce the spatial consistency, we sample the SGS stresses according to

$$\tau(x, t) = \sigma_\theta(u(x, t))Z(t) + \mu_\theta(u(x, t)), \quad Z(t) \sim \mathcal{N}(0, 1). \tag{11}$$

We use the reparametrization trick to sample from Gaussian distribution where $Z(t)$ is sampled from standard Gaussian distribution. As this sampler only depends on t rather than x, the random noise of the generative model will be consistent along all the spatial points at the same time step while only change according to time. For the temporal consistency it suffices to change $Z(t)$ to $Z(x)$ and for spatial-temporal consistency it suffices to choose a random Z for the whole simulation.

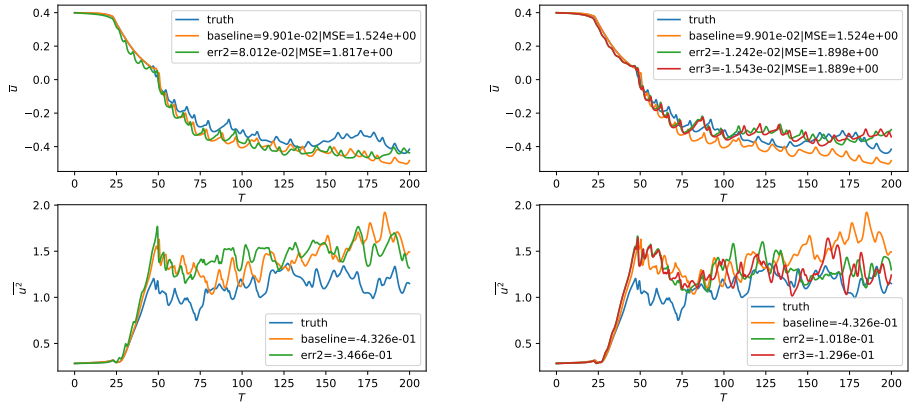
After this discussion, we can present our experiments results on the KS equation. In summary, we perform five simulations:

1. Fine-grid simulation without any SGS model: ground truth.
2. Coarse-grid (4x coarser) simulation without any SGS model (baseline).
3. Coarse-grid simulation with regression-based SGS model (regression).
4. Coarse-grid simulation with spatial-temporally consistent generative SGS model (gaussian, fix).
5. Coarse-grid simulation with only spatially consistent generative SGS model (gaussian, sample).

The name in the brackets refers to their column in table 2. The simulation results are shown in fig. 2. For both figures, the simulations on fine grid and coarse grid without any SGS model are given by the blue (truth) and orange (baseline) curve respectively. In the left figure, the coarse grid simulation with regression-based SGS model is the green (err2) line. In the right figure, the coarse grid simulation with generative SGS model and spatial-temporal consistency is the green (err2) line while with only spatial consistency is the red (err3) line. The number in the legend refers to the error of the first and second moment statistics averaged over time and space. We also calculate the pointwise error in eq. (7) denoted by MSE.

It can be observed from fig. 2a that for both moments the regression-based SGS model has marginal improvements over the baseline. On the other hand, for generative SGS modeling shown in fig. 2b, both generative SGS model has much greater improvements over the baseline than the regression model and with spatial-temporal consistency the improvement is slightly greater. This results are also stated in the table table 2.

We conduct complete numerical experiments over different input features and SGS models and report their MSE of the trajectory as well as the error of the first second moments. The results are summarized in table 2.



(a) Regression based the SGS modeling.

(b) Generative SGS modeling.

Figure 2: Comparison of the a-posteriori error over different approaches to SGS modeling.

4 Conclusion

In this work, aiming to understand the mismatch of a-priori and a-posteriori error in SGS modeling, we discover two issues, data imbalance and multi-valuedness, by investigating the dataset with evidences given by visualizations. We attribute the substantial training error of the SGS models to the multi-valuedness. To resolve this, we propose a generative SGS model improving the single-value feature of the classical regression-based model and demonstrate its effectiveness to reduce the a-posteriori error on the KS equation.

References

Patrick J Blonigan and Qiqi Wang. Least squares shadowing sensitivity analysis of a modified kuramoto–sivashinsky equation. *Chaos, Solitons & Fractals*, 64:16–25, 2014.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

Myoungkyu Lee and Robert D Moser. Direct numerical simulation of turbulent channel flow up to. *Journal of fluid mechanics*, 774:395–415, 2015.

Table 2: Comparison of the a-priori and a-posteriori error of the method for SGS modeling of the KS equation. For a-priori error, we use the relative MSE for regression-based method while for generative modeling we use the expectation value of the log likelihood function. For a-posteriori error, we use both the trajectorywise error and the first and second momemts. Four rows correspond to the last four simulations in 3.2.

	baseline	regression	gaussian, fix	gaussian sample
a-priori error	N/A	0.976	-2.173	-2.173
$\int (u-u_0)^2 dxdt$	1.524	2.036	1.720	1.597
$\overline{u} - \overline{u_0}$	9.901E-02	1.011E-01	3.870E-02	3.214E-02
$\overline{u^2} - \overline{u_0^2}$	-4.326E-01	-4.241E-01	-1.895E-01	-6.577E-02
a-priori error	N/A	0.987	-2.583	-2.583
$\int (u-u_0)^2 dxdt$	1.524	1.817	1.898	1.889
$\overline{u} - \overline{u_0}$	9.901E-02	8.012E-02	-1.242E-02	-1.543E-02
$\overline{u^2} - \overline{u_0^2}$	-4.326E-01	-3.466E-01	-1.018E-01	-1.296E-01

Romit Maulik and Omer San. A neural network approach for the blind deconvolution of turbulent flows. *Journal of Fluid Mechanics*, 831:151–181, 2017.

Romit Maulik, Omer San, Adil Rasheed, and Prakash Vedula. Subgrid modelling for two-dimensional turbulence using neural networks. *Journal of Fluid Mechanics*, 858:122–144, 2019.

Jonghwan Park and Haecheon Choi. Toward neural-network-based large eddy simulation: Application to turbulent channel flow. *Journal of Fluid Mechanics*, 914:A16, 2021.

Benjamin Sanderse, Panos Stinis, Romit Maulik, and Shady E Ahmed. Scientific machine learning for closure models in multiscale problems: A review. *arXiv preprint arXiv:2403.02913*, 2024.

Chuwei Wang, Julius Berner, Zongyi Li, Di Zhou, Jiayun Wang, Jane Bae, and Anima Anandkumar. Beyond closure models: Learning chaotic-systems via physics-informed neural operators. *arXiv preprint arXiv:2408.05177*, 2024.

Henry G Weller, Gavin Tabor, Hrvoje Jasak, and Christer Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in physics*, 12(6): 620–631, 1998.

Jiaxi Zhao and Qianxiao Li. Mitigating distribution shift in machine learning-augmented hybrid simulation. *arXiv preprint arXiv:2401.09259*, 2024.

A Experiments details

A.1 Dataset generation

In this section, we discuss the procedure to generate the dataset of the SGS modeling from high fidelity simulation.

To generate the dataset for NS equations, we use OpenFOAM (Weller et al., 1998) to perform the simulation over a periodic channel with upper and bottom wall. We follow the existing direct numerical simulation literature (Lee & Moser, 2015) to set up the numerical experiments and verify our simulation results with the literature by various statistics. Throughout the paper, we use x, y, z to denote the streamwise, wall-normal, and spanwise direction respectively. Moreover, we are mostly interested in the behavior of several statistics versus the wall-normal direction y and in these cases we will take average over both x, z directions.

Given the simulation results and recall the definition of the SGS stress in eq. (4), we perform discrete convolution to obtain the stresses from the discrete measurements of the velocity. Some typical choices are the box filter, Gaussian filter, spectral filter, etc. Throughout the paper, we fix the box filter as the filter operator. Specifically, for 3D simulation with a box filter of length L, the stress can be calculated as

$$\tau_{ij}(x_0, y_0, z_0) = \frac{1}{L^3} \sum_{x,y,z} \widetilde{u}_i(x, y, z)\widetilde{u}_j(x, y, z) - \frac{1}{L^6} \left(\sum_{x,y,z} \widetilde{u}_i(x, y, z) \right) \left(\sum_{x,y,z} \widetilde{u}_j(x, y, z) \right) \tag{12}$$

where the summation is taken over the cube centered at (x_0, y_0, z_0) with length L, i.e. $x \in [x_0 - L/2, x_0 + L/2], y \in [y_0 - L/2, y_0 + L/2], z \in [z_0 - L/2, z_0 + L/2]$. In our case, we filter merely along the x, z direction and retains all the information of the wall-normal direction. The box filter we used has side length 4 times the grid size. Besides the filtered velocity, we can also obtain the filtered shear stresses , rate of strain tensor and wall unit and use them as the input features of our SGS models. The dataset will consist of data pair of input features and SGS stresses at each spatial-temporal point.

The KS equation and the boundary condition we considered is given by

$$\begin{aligned} u_t &= -(c + u)u_x - uu_x - u_{xx} - \nu u_{xxxx}, \\ u(0, t) &= u(L, t) = 0, \\ u_x(0, t) &= u_x(L, t) = 0, \forall t. \end{aligned} \tag{13}$$

To solve the equation, the space is discretized via finite difference (mostly central difference) and we use Crank-Nicolson scheme to solve the time evolution. Both the boundary condition and the numerical scheme is consistent with that of Blonigan & Wang (2014). The SGS modeling dataset can be generated similarly as that of NS equations with a box filter of size 4. This corresponds to the ratio of the fine and coarse mesh resolutions.

A.2 Visualizing the multi-valuedness in multi-dimension

In the main text, we visualize the correlation between the input feature and the SGS stresses in the KS equation by the scattering plot fig. 1 of two scalar and demonstrate the issue of multi-valuedness in that case. Visualization of the multi-valuedness for the case of NS equations is not straightforward as that of the KS equation as both the input features, e.g. filtered velocity, rate of strain tensor and the output features SGS stresses are not 1D. Two ingredients are required for the visualization. Firstly, we have to perform dimension reduction on the 9D SGS stresses so that we can plot them on the plane. Specifically, we perform PCA on the set of 9D vector SGS stresses and project them onto the first two principle components which account for more than 90% of variance and thus contain enough information. Next, inspired by the notion of continuity for mapping that neighboring inputs should have neighboring output.

We use k-nearest neighbor algorithm to find a cluster of neighboring input features, visualize their corresponding outputs using dimension reduction technique, and check if this outputs are neighbors to each other. If we can detect some clustering patterns in the outputs plot, this can be an evidence to the multi-valuedness of the SGS modeling. The pseudocode of this visualization algorithm is summarized in algorithm 1.

Algorithm 1 Visualizing the multivalue phenomena in multidimensional datasets.

Input: Dataset $\{\mathbf{u}^{(n)}, \tau^{(n)}\}$, an array of number of neighbor;

1: Choose a data pair $(\mathbf{u}^{(0)}, \tau^{(0)})$ randomly from the dataset;
2: Perform dimension reduction, e.g. PCA, tSNE, UMAP on the stress data $\{\tau^{(n)}\}$ and project them onto a two-dimensional space.
3: **for** n in n_array **do**
4: Run k-nearest neighbor algorithm to find the n nearest neighbor of $(\mathbf{u}_0, \tau_0)$ in the whole dataset;
5: Visualize the stresses corresponding to these nearest neighbors using the projection obtained from dimension reduction.
6: **end for**
7: Seek the cluster patterns in the low-dimensional visualization of the stresses plot as the number of nearest neighbor decreases.