Deep Learning Modeling of Sub-grid Scale Turbulence in Large-Eddy Simulations

Chen Cao, Ling Lin, Pit Kauffmann, Cody Zupnick

cc4207@columbia.edu, ll2948@columbia.edu, pk2573@columbia.edu, cjz2104@columbia.edu

Columbia University, Fu Foundation School of Engineering and Applied Science

Turbulence modeling has long been one of the most challenging problems in engineering and physics. This report discusses a relatively novel approach of modeling sub-grid scale turbulence characteristics resulting from spatial averaging in Large-Eddy Simulations. We apply deep learning techniques to model sheer stress and heat flux across space and time, with the goal of finding a function that generalizes well across differing initial conditions. We introduce a separate filter operation to pre-process our data for a deep network and introduce time-dependencies via recurrent neural networks. Our results compare well to those of recent research using similar approaches and show a significant improvement over traditional methods, both in terms of computational efficiency and predictive accuracy. We obtain correlations between true and predicted values of up to 0.9 for both sheer stress and heat flux, and introduce models that combine different initial conditions to obtain superior robustness.

0 INTRODUCTION

Turbulence modeling remains one of the biggest challenges in engineering today, and has impacts ranging from aerodynamics to climate modeling. A plane cannot fly without jet propulsion and your car consumes the amount of fuel it does partly due to its aerodynamic properties. This report focuses on atmospheric turbulence related to meteorology in the context of cloud pattern prediction, which can significantly improve weather and climate predictions by for example resulting in a better understanding of how a cloud's density, and hence its ability to reflect and absorb CO_2 changes.

Given the difficulty and high cost of observing turbulence in a natural environment, most turbulence models include some form of simulation. Among the most popular methods are Direct Numerical Simulations (DNS), and Large-Eddy Simulations, first proposed in 1963 by Joseph Smagorinsky [8] and first explored in 1970 by Deardorff [2].

The governing equations of these types of simulations are the Navier-Stokes equations, which represent Newton's second law applied to the x, y and z-direction of Newtonian fluids in motion:

$$\rho\left(\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}\right)=\rho g_x-\frac{\partial P}{\partial x}+\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right) \quad (1)$$

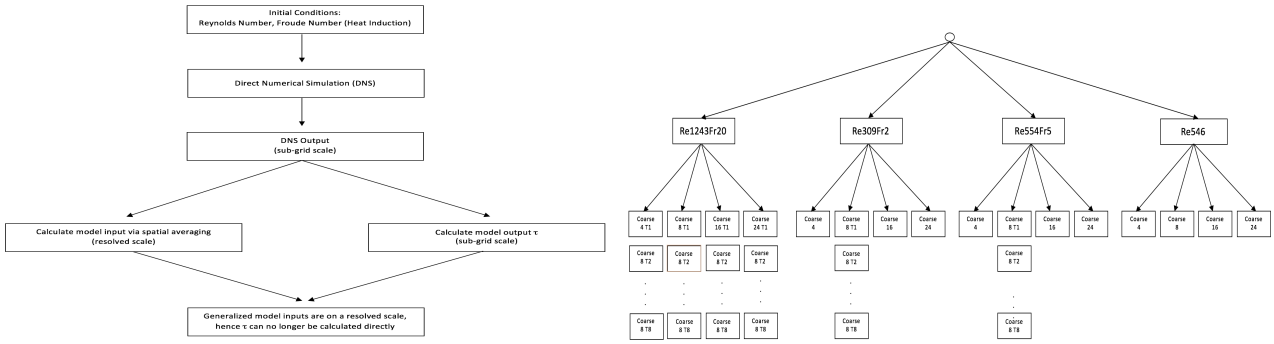
$$\rho\left(\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}\right)=\rho g_y-\frac{\partial P}{\partial y}+\mu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 v}{\partial z^2}\right) \quad (2)$$

$$\rho\left(\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z}\right)=\rho g_z-\frac{\partial P}{\partial z}+\mu\left(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{\partial^2 w}{\partial z^2}\right) \quad (3)$$

Hence, the first equation indicates that the sum of forces in the x direction has to equal the mass times the acceleration in the x direction, where the forces to be considered are gravity, forces resulting from differences in pressure and forces due to fluid viscosity (i.e. the gradual deformation produced by stress).
Both simulation-based approaches mentioned above use these equations as follows:

- Direct Numerical Simulations (DNS) analytically solve the Navier-Stokes equation across all temporal and spatial scales. The main issue arises because current computing power is not sufficient to conduct a DNS on large scales.
- Large Eddy Simulations (LES) are more computationally efficient, by attempting to solve the Navier-Stokes equations on large scales only and subsequently model the turbulence behavior on smaller scales. This is achieved via low-passed spacial filtering (i.e. a convolution kernel $G_\Delta(x)$ of size Δ is applied), a method used to separate large (resolved scale) and small (sub-grid scale) eddys. Applying the kernel will remove sub-grid scale information and transforms the Navier-Stokes equations by adding a stress term τ^Δ that needs to be modeled.

The goal of our research was to identify new, efficient ways to model sub-grid scale turbulence via deep learning by using the pseudo-resolved scale outputs from an iso-



(a) Data Generating Process

(b) Data Category Overview

Fig. 1: (a) Data generating process showing the output of DNS with differing initial conditions is averaged to obtain our model input and output data. (b) Tree-based view of our dataset categories.

lated DNS. Specifically, we want to apply Convolutional, Dense and Recurrent neural networks in order to model small-scale turbulence across space and time using non-linear regression analysis.

1 RELATED WORK & CONTRIBUTIONS

The main challenge in turbulence research has been understanding the relationship between small and large scale motions within a given structure, dynamic and statistic. In the past decades, one property that people have paid a lot of attention to is scale invariance, which indicates that even at different scales, certain features of a flow exhibit the same properties.

In most cases, calculating turbulent flows in complete detail via DNS (on all temporal and spatial scales) is not practical, because the range of length an time scales is so immense that the amount of data needed to be processed exceeds the computing power capacity of any existing super-computer. Consequently, most research efforts have been focused on Large-Eddy-Simulations for sub-grid scale turbulence modeling. While LES are more computationally efficient, they also allow for the addition of scale-dependent variable (due to spatial filtering), whereas traditional models have often included feature matrices of spatial velocity fields (such as gradients).

One of the first applications came in 1979, when Clark [1] used a calculation of periodic homogeneous isotropic turbulence to simulate the experimental decay of grid turbulence, by computing the large eddy fields and averaging the sub-grid scale turbulence from the simulation's modified equations. They found that the large-scale field can be accurately modeled but the sub-grid scale Reynolds stresses were only moderately well modeled.

In 1982, Moin and Tim [7] from Stanford University published their work, showing LES can be used effectively. In 2000, Meneveau and Katz [6] described several sub-grid Scale models using scale invariance concepts, i.e. the Smagorinsky model, the dynamic model, and the mixed similarity model. They focused on modeling the sub-grid scale stress tensors τ. For these models, coefficients for τ need to be modeled. Finally, in 2017, Gamahara and Hattori [3] from Tohoku first introduced a way of using artificial neural networks to find a new model of sub-grid scale stresses in Large-Eddy Simulations. Without assumptions about the form of the relationship, a multilayer neural network is set up to establish a functional relation between the grid-scale flow field and the SGS stress. Data fed into the ANN is provided by a numerical simulation of turbulent channel flows. The results showed that an ANN can establish a predictive model with capabilities similar to those of a gradient model. In this project, we will build on the results from Gamahara and Hattori's research and introduce a filter-based pre-processing technique, construct additional modeling with input datasets using multiple initial conditions, and introduce time-dependent models.

2 DATA DISCUSSION

2.1 DATA GENERATING PROCESS

Our datasets consist of 3-dimensional spatial cubes and were generated using a high resolution DNS with differing initial conditions over a small spatial area. The sub-grid scale DNS output was then used to (1) calculate the model inputs via averaging over regions of differing sizes and (2) calculate the model outputs. With the DNS output being on a sub-grid scale, the spatial averaging was done in order to achieve a good proxy for the generalized input to the model, which will be on a resolved scale. Our goal was then to identify a function that maps input to output, since the true input will be on a resolved scale and will no longer allow us to calculate the output directly (see Fig. 1-a).

2.2 DATA OVERVIEW

A summary of our dataset categories can be found in Fig. 1-b. These categories can be understood as follows:

Layer 1: DNS Initial Conditions. Each node is labeled Re-x-Fr-y, where x is the Reynolds Number and y is the Froude Number.

- The Reynolds Number is a dimensionless measure of turbulence. The higher the number the more turbulent the flow.
- The Froude Number is a dimensionless, relative measure of forces, defined as the ratio of inertial forces (due to particles of fluid resisting change in momentum) to external fields (which is often just gravity)

Loosely speaking, the Froude number quantifies the addition of heat conduction from underneath as an initial condition. Such heat conduction may, for example, model stratified turbulence in the upper troposphere (i.e. the earth's lowest atmospheric layer). The last modeling setup (Re546) does not include any conductive heating and hence a Froude Number is not necessary. Note that the selected Reynolds and Froude Numbers are standard practice in turbulence modeling and are set by the researchers ([5], [4])

Layer 2: Averaging Range. For each Reynolds Number - Froude Number combination, we have different categories identified by coarse-z, where z is either 4, 8, 16 or 24. These numbers represent the grid-size over which the DNS outputs are averaged. As mentioned above, the DNS is averaged over a range of size (2z+1)x(2z+1)x(2z+1) in order to better mimic the generalized input to the model later on. Please note that for several data categories, we have data for 8 different time steps, T1 - T8.

Layer 3 (not pictured): Datasets. Within each Re-x-Fr-y coarse-z category, we have one sample of each of the following 3-D boxes for each time step:

Variable	Type	Description of Box Entries
u	Input	velocity in the x-direction
v	Input	velocity in the y-direction
w	Input	velocity in the z-direction
tke	Input	total kinetic energy
θ	Input	temperature
p	Input	pressure
h_{xyz}	Output	heat flux
τ^{Δ}_{xyz}	Output	stress due to low-pass filtering

Due to the spatial averaging of the DNS output over regions of differing size, the four coarse-z data categories have input and output cubes of different sizes. They are as follows:

Coarse	Grid Size	Input / Output Size
4	(9x9x9)	(296, 196, 153)
8	(17x17x17)	(146, 96, 72)
16	(33x33x33)	(71, 46, 36)
24	(49x49x49)	(46, 30, 23)

2.3 EXPLORATORY ANALYSIS

By plotting our input boxes (see Fig. 2 a-c), we observe that they all show a similar pattern. Turbulence is highest in the low layers along the z-axis, and then gradually dissipates as one moves up. This can be partially explained by the vanishing impact of the heat conduction from underneath, the effect of which diminishes as we move along the z-direction. Additionally, you may notice that the last few layers along the z-direction show essentially no turbulence. These are laminar layers added by the researchers during the DNS and were removed prior to modeling in order to avoid any distortions. Other input datasets and all output datasets show highly similar characteristics.

We can further see (fig 2. d-f) that most inputs are typically normally distributed with very few outliers, little skew and means close to 0 (v, w) and 1 (u). However, note the differing means and levels of skewness and kurtosis for differing initial conditions as well as the fairly oddly shaped distribution of Re546-coarse8-u. This can partially be explained by the absence of heat conduction, which causes the x-direction velocities to lose normality and "drag out". These anomalies will heavily impact our model accuracy later on when attempting to generalize our models to different initial conditions and can be mitigated with diversified training on multiple conditions.

3 MODELING APPROACH

The underlying assumption is that there exist some functional relationships between τ and u, v, w, tke, θ and p. We try to capture this relationship using a multi-layer Neural Network model. By the Universal Approximation Theorem, any continuous function can be approximated by a simple feed-forward network, in theory. As the inputs of our model (u, v, w) and its outputs (τ_{12}, τ_{13} and τ_{23}) exhibit an analytical relationship, this theorem provides valid support for our deep learning model.

3.1 DATA PROCESSING

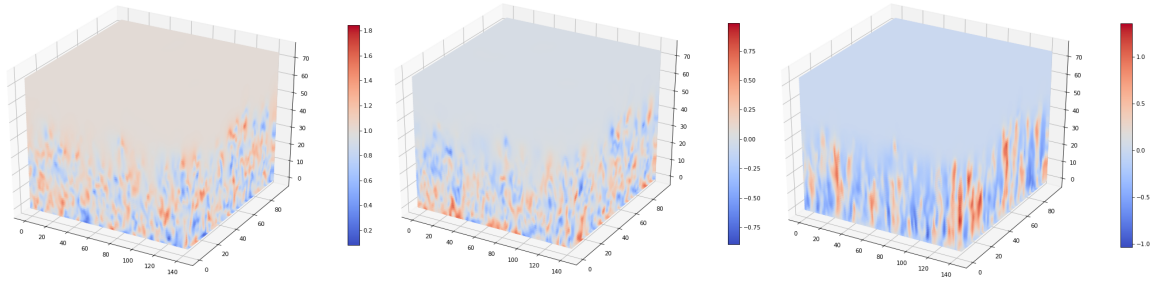
We have applied several processing steps to prepare the data to be fed into our models.

Step 1: We combined all input boxes (u,v,w,tke,θ,p) into a single 4-D tensor.

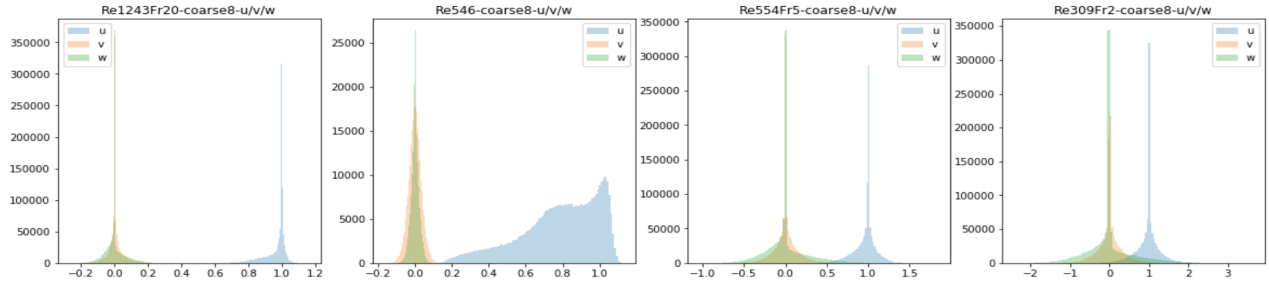
Step 2: In order to break the larger inputs into samples, we applied a technique similar to the idea behind filtering in a CNN. The assumption is that we do not need the entire input dataset to predict each τ; rather each τ can be predicted by a small box of size dxdxd surrounding the corresponding value in the input set. Those small boxes then constitute our input dataset. Please note that this method removes all dependencies that might exist beyond the small boxes of size dxdxd.

Step 3: We cut off all datasets at about 75% along the z-dimension. As mentioned above, the researchers have added a laminar layer on top of all boxes, which we remove to avoid any distortion.

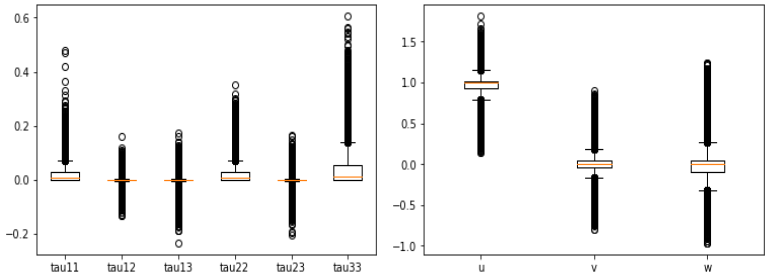
Step 4: We limited our analysis to focus only on three of the six τ datasets, namely τ_{12}, τ_{13} and τ_{23}. Those represent the most important stresses resulting from the spatial averaging technique that is applied to the DNS output, and exhibit the most instability.



(a) $u_{xyz}(Re1243Fr20 - coarse8)$ (b) $v_{xyz}(Re1243Fr20 - coarse8)$ (c) $w_{xyz}(Re1243Fr20 - coarse8)$



(d) Input distributions for differing initial conditions



(e) Outputs (Re1243Fr20-coarse8) (f) Inputs (Re1243Fr20-coarse8)

Fig. 2: (a) - (c) Select input matrices. (d) Distribution of u, v and w for all 4 initial conditions (averaging range coarse8). (e) - (f) boxplots of select inputs and all 6 sheer stresses

3.2 MODEL SET-UP

We have tried different types of deep learning models: DNN (base model), CNN (to capture spatial dependencies) and LSTM (to model time-dependencies). Initially, we trained independent models for each sheer stress and heat flux. Comparing the results for each model, we chose the DNN as our final model, due to its simplicity and accuracy. As for the training process, we have considered different input / output combinations, in order to determine which would be most generalizable.

- Train on Re1243Fr20-coarse8 (base model)
- Train on Re1243Fr20-coarse4, Re1243Fr20-coarse8, Re1243Fr20-coarse16 and Re1243Fr20-coarse24 to isolate the effects of the averaging range
- Train on Re309Fr2-coarse8 and Re1243Fr20-coarse8 to isolate the effects of the initial conditions. Here Re309Fr2 shows the least amount of turbulence while Re1243Fr20 shows the most turbulence among all our datasets
- Train on Re309Fr2-coarse8 and Re1243Fr20-coarse8, both at time steps T1 and T2 to isolate the effects of temporal information

3.3 PARAMETER TUNING

We started with a basic set of assumptions around the data and model.

- Input data: Re1243Fr20-coarse8 u,v,w. This dataset was used because it can be viewed as the most ”turbulent”, which will hopefully allow our model to detect meaningful features that generalize well.
- Input data box size: 7x7x7x3, based on previous research
- Loss Function: mean-squared error
- Performance metric: correlation between predicted and actual τ.
- Normalization technique: max normalization (i.e. divide by the max)

Subsequently, we tried different variations of the above (such as different box sizes or loss functions) to understand if our initial assumptions can be improved upon.

1. **Decision 1: Max normalization vs. Standard normalization.** We wanted to determine whether different normalization methods would give us significantly different results. In order to achieve this, We ran the DNN using the basic assumptions mentioned at the beginning of this section, and varying the normalization technique between our default max normalization as well as mean-variance standardization. Tracking correlation as the metric, we obtained the following results:

Normalization Method	τ_{12}	τ_{13}	τ_{23}
Max	0.7524	0.8231	0.8299
Standard	0.8443	0.8963	0.8755

Based on the results, we moved forward with the mean-variance standardization technique, as it leads to higher correlations across the board. Our analysis of the data has shown that the large improvement in the correlation metric for τ_{12} can be explained by the fact that τ_{12} is particularly volatile. Hence, dividing by the standard deviation during the normalization process removes some of this variability.

2. **Decision 2: Adding independent variables tke, θ, p.** As mentioned above, our base assumption uses only the velocity fields as input to our model. However, given that we have a lot more data available, we wanted to determine whether adding any of the other 3 potential inputs (tke, θ and p) has any significant impact. Hence, we ran the DNN again, using mean-variance standardization and varying the input data by adding the respective variables. The correlations are shown below:

Input Variables	τ_{12}	τ_{13}	τ_{23}
u,v,w	0.8501	0.8947	0.8750
u,v,w,tke	0.8455	0.8963	0.8969
u,v,w,θ	0.8486	0.8987	0.8763
u,v,w,p	0.8488	0.8979	0.8727

Given there is no significant improvement resulting from adding the additional independent variables, we decided to move forward with the smallest dataset, composed of u,v and w only.

3. **Decision 3: MSE vs. Correlation loss function.** Given that we are attempting to maximize the correlation, we attempted to build a custom loss function based on the correlation coefficient. We ran the DNN model, using mean-variance standardization on the inputs (u,v and w) and varied the loss function between mse and our custom loss (which is 1 - corr, such that the loss can be minimized correctly). The results we obtained are shown below.

Loss Function	τ_{12}	τ_{13}	τ_{23}
MSE	0.8470	0.8958	0.8758
Correlation	0.8477	0.8962	0.8786

We can see form the above that there is essentially no difference between the two loss methods. However, it would be difficult to justify using correlation as a custom loss function because of a difficult-to-interpret gradient as well as no method for preventing the absolute difference between the predicted and the true τ from increasing too much. Furthermore, since the correlation is inherently only meaningful on batches of data (i.e. correlation for one input and its output is not defined), correlation loss would make our calculations dependent on the chosen batch size. This will lead to issues during gradient descend, as the gradients of each sample in a batch are no longer independent and relate to batch size, making them susceptible to the vanishing/exploding gradient problem. For those reasons, we decided to move forward with the mse loss.

4. **Decision 4: Use training samples with different dimensions.** Lastly, we wanted to understand whether different sample dimensions would be impactful. Specifically, we wanted to see if each tau can be better predicted using different sized boxes around its corresponding input value. Consequently, we ran a DNN one last time, using mean-variance standardization on u,v and w, mse loss and varied the shape of each input box. The resulting correlations for our box sizes are shown below:

Input Shape	τ_{12}	τ_{13}	τ_{23}
7x7x7x3	0.8470	0.8950	0.8747
11x7x5x3	0.8469	0.8933	0.8692
15x11x3x3	0.8356	0.8952	0.8703
3x3x3	0.8281	0.8532	0.8415

Note that the different box sizes were chosen mainly in order to identify whether considering a greater region along the x-dimension and considering a smaller region along the z-dimension would have a significant impact on correlation. The main idea was that since turbulence is generally dissipating along the z-axis, its impact can be minimized by avoiding to select boxes with too large of a range along the z-axis. Similarly to the case of the input variables, we decided to move forward with the smallest box size that does not negatively impact correlation, namely the 7x7x7x3 box.

After evaluating the parameter search discussed above, we constructed our final model as follows:

- Input data: u,v,w.
- Input data box size: 7x7x7x3
- Loss Function: MSE. As the purpose is to maximize the correlation between the predicted values and the ground

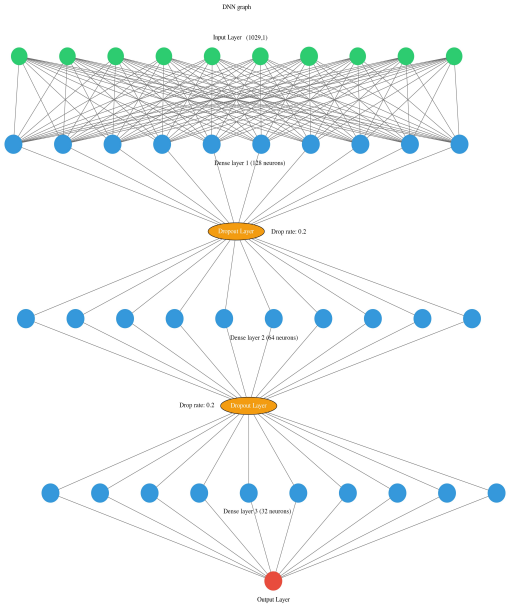


Fig. 3: Final DNN Architecture

truth, the MSE presents a natural fit and its gradient will have a simple form, which allows for quick convergence.
- Performance metric: correlation between predicted and actual τ
- Normalization technique: mean-variance standardization

Our final DNN architecture can be summarized as follows:

Layer	
Input	(784896, 7x7x7x3)
Layer 1	Flatten
Layer 2	Dense (nodes: 128, act: relu)
Layer 3	Dropout (rate: 0.2)
Layer 4	Dense (nodes: 64, act: relu)
Layer 5	Dropout (rate: 0.2)
Layer 6	Dense (nodes: 32, act: relu)
Layer 7	Dense (nodes: 1)

3.4 EXPERIMNETAL MODELING

Apart from the base single-input single-output DNN model discussed above, we also tried the following:

- Modify the DNN to become single-input, multi-output and predict all sheer stresses and heat flux concurrently. This was done to take advantage of weight sharing that could improve generalization. The resulting predicted correlations for a selection of sheer stresses are shown below and do not indicate an improvement in generalizability compared to the base model:

Dataset	Base Model	Multi-Out
Re1243Fr20-coarse8-τ_{12}	0.8674	0.8282
Re309Fr2-coarse8-τ_{12}	0.6965	0.6950
Re546-coarse8-τ_{12}	0.5666	0.4818
Re554Fr5-coarse8-τ_{12}	0.7259	0.7235

- Modify the DNN to become multi-input, multi-output to predict all sheer stresses and heat flux simultaneously using input data across different initial conditions. The results below compare our base DNN to a model that predicts all sheer stresses and heat flux using the least and the most turbulent datasets as input (as discussed in Section 3.2). Once again the results are rather mixed, indicating some improvements for some initial conditions, but worse accuracy for others.:

Dataset	Base Model	Multi-In-Out
Re1243Fr20-coarse8-τ_{12}	0.8674	0.8222
Re309Fr2-coarse8-τ_{12}	0.6965	0.7905
Re546-coarse8-τ_{12}	0.5666	0.4893
Re554Fr5-coarse8-τ_{12}	0.7259	0.7909

- Use a CNN to incorporate all spatial dependencies of the data. Our entire DNN architecture, plus pre-processing can be replicated using a CNN, yet we do not have enough data to generate accurate predictions, as each full-sized input box acts as 1 sample in this case. Given the limited number of sample we have, model parameters can only be updated a few times.
- Use an LSTM model with time-distributed layers to incorporate temporal dependencies and predict the results for multiple time periods for each individual τ. Similarly to the CNN discussed above, the 8 time steps of available data are not enough to train and accurate model that generalizes well. The training progression of our correlation metric for all three taus in the Re1243Fr20-coarse8 category are summarized below:

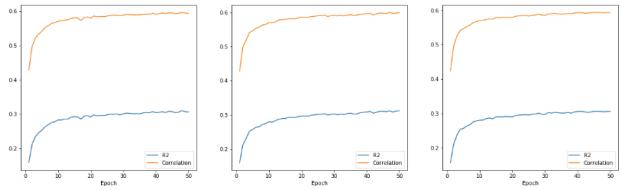


Fig. 4: Correlation and R^2 metric progression by epoch for the LSTM ($\tau_{12}, \tau_{13}, \tau_{23}$)

The above models remain mainly experimental for the purposes of this project and will require a more in-depth, formalized analysis going forward, using more input data.

4 RESULTS

4.1 PREDICTION OF SHEER STRESS τ

After constructing our optimal model, the goal was to train a model that generalizes well enough across different dataset categories (i.e. different initial conditions/averaging ranges). As mentioned above, we trained the model on different combination of datasets to maximize its representative abilities. We then used it to make predictions on various other dataset categories, such as all other averaging ranges within Re1243Fr20, as well as all other coarse8 datasets available (Re554Fr5-Coarse8, Re546-Coarse8 and Re309Fr2-Coarse8). The summarized results are discussed below:

1. **Scenario 1: Train single-output models on Re1243Fr20-coarse8.** Predicted vs. actual correlations for different averaging ranges and initial conditions at time step T1 are shown below:

Dataset Category	τ_{12}	τ_{13}	τ_{23}
Re1243Fr20-coarse16	0.7248	0.8304	0.7897
Re1243Fr20-coarse24	0.6551	0.7887	0.7218
Re1243Fr20-coarse8	0.8674	0.9076	0.8862
Re309Fr2-coarse16	0.5418	0.6788	0.7229
Re309Fr2-coarse24	0.4263	0.5738	0.6331
Re309Fr2-coarse8	0.6965	0.7721	0.8014
Re546-coarse16	0.4326	0.6012	0.5004
Re546-coarse24	0.3939	0.6429	0.4524
Re546-coarse8	0.5666	0.6008	0.6397
Re554Fr5-coarse16	0.6025	0.7115	0.7356
Re554Fr5-coarse24	0.5000	0.6301	0.6594
Re554Fr5-coarse8	0.7259	0.7883	0.8087

Predicted vs. actual distributions for select averaging ranges and initial conditions are show in Figure 6. We can see that, although the distributions are mostly very normal, the level of kurtosis and skew varies significantly. For example, datasets in the Re546 category (which, as you recall has no heat conduction), show much fatter tails. This insight was the main reason for the multi-input models described above that consider several initial conditions simultaneously. A very interesting fact can be observed in Figure 5, where we plot the correlations for each τ across each layer of the z-direction. The model's accuracy significantly drops towards the last few layers. Remember that those layers show very little turbulence as it dissipates along the vertical axis. This indicates that we can generalize well for high-turbulent regions, but fail to capture some of the outliers in the dissipating regions.

2. **Scenario 2: Train single-output models on Re1243Fr20-coarse8 and Re309Fr2-coarse8.** To remedy some of the complexity and variability added through initial conditions, we trained these models on two distinct initial conditions. Predicted vs. actual correlations for different averaging ranges and initial conditions at time step T1 are shown below:

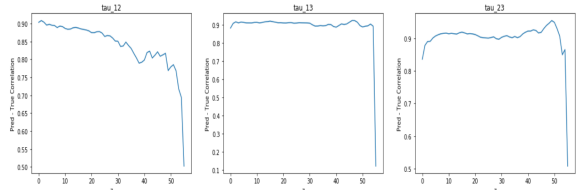


Fig. 5: True vs. predicted τ correlations for each layer along the z-dimension

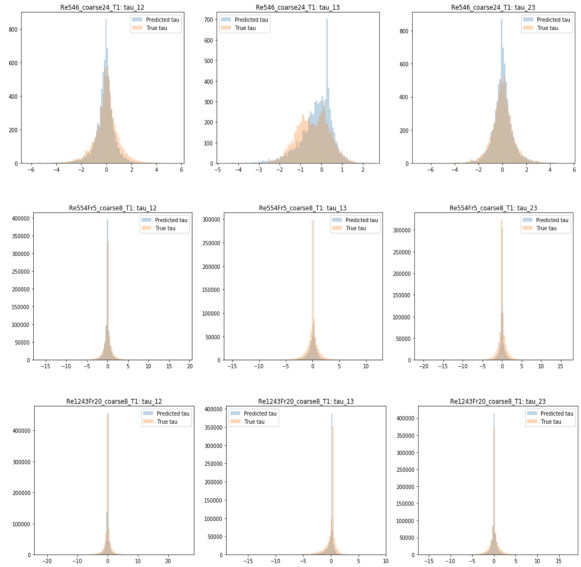


Fig. 6: Select datasets showing true vs. predicted τ distributions for single-output model trained on Re1243Fr20-coarse8

Dataset Category	τ_{12}	τ_{13}	τ_{23}
Re1243Fr20-coarse16	0.7529	0.855	0.8115
Re1243Fr20-coarse24	0.6944	0.8166	0.7459
Re1243Fr20-coarse8	0.8576	0.9017	0.8877
Re309Fr2-coarse16	0.6641	0.7571	0.7762
Re309Fr2-coarse24	0.5370	0.6508	0.6873
Re309Fr2-coarse8	0.8409	0.8762	0.8892
Re546-coarse16	0.4050	0.6281	0.5437
Re546-coarse24	0.4141	0.6174	0.4846
Re546-coarse8	0.4880	0.6172	0.6896
Re554Fr5-coarse16	0.6992	0.7772	0.7855
Re554Fr5-coarse24	0.5959	0.6997	0.7093
Re554Fr5-coarse8	0.8186	0.8493	0.8664

Notice how the results significantly improve for the Re309Fr2 datasets. The intuition is that by training on the most and least turbulent datasets, we are able to model a function that generalizes well to turbulence levels in between. Once again, we plot the distribution of select true vs. predicted τ in Figure 7, noticing that some of the spikes around the means have been reduced.

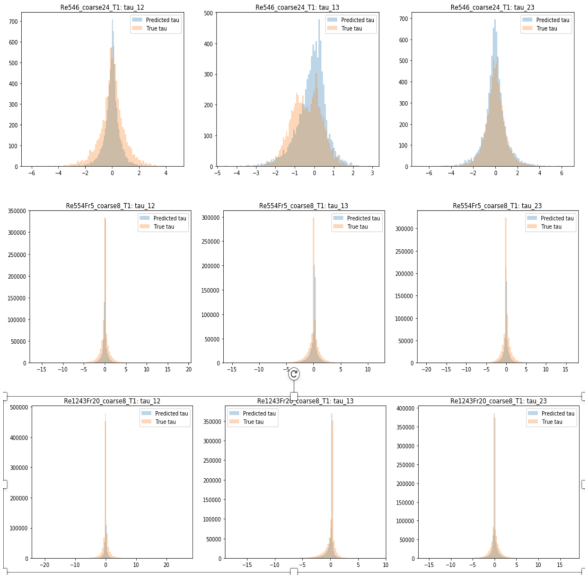


Fig. 7: Select dataset categories showing true vs. predicted τ distributions for single-output model trained on Re1243Fr20-coarse8 and Re309Fr2-coarse8

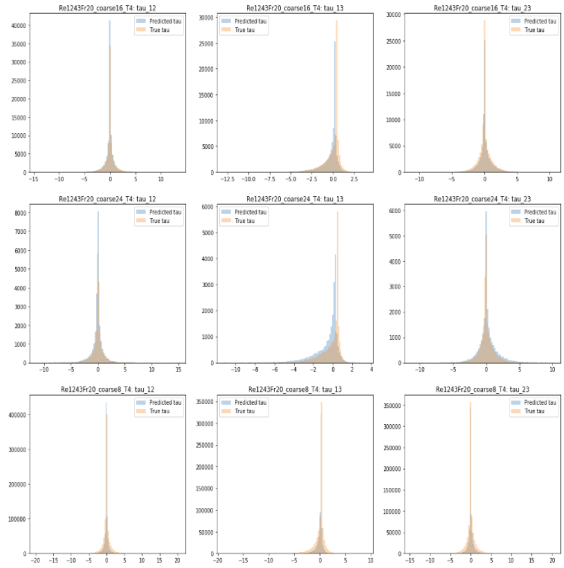


Fig. 8: Select dataset categories showing true vs. predicted τ distributions for single-output model trained on Re1243Fr20-coarse8 and Re309Fr2-coarse8 T1 and T2

3. **Scenario 3: Train single-output models on Re1243Fr20-coarse8 and Re309Fr2-coarse8 at T1 and T2.** We wanted to understand what impact the time dimensionality would have without using a recurrent model. We consequentially trained a model on different initial conditions and different time steps without giving the model any information that time was a factor. The idea was that adding data from different time steps might allow us to make predictions for different time steps as well. We show predicted vs. actual correlations for different averaging ranges and initial conditions at a different time step T4 below. Please note that the predictions are limited to the dataset categories for which data at time step T4 was available (refer to Fig. 1-b for an overview):

Dataset Category	τ_{12}	τ_{13}	τ_{23}
Re1243Fr20-coarse16	0.7764	0.8489	0.8213
Re1243Fr20-coarse24	0.7185	0.8077	0.7740
Re1243Fr20-coarse8	0.8400	0.8807	0.8741
Re309Fr2-coarse8	0.8255	0.8722	0.8743

We can quickly observe that adding time steps to our input data without introducing any recurrences does seem to provide us with some ability to generalize to other time steps, with all correlations remaining in the same range as those for the same time step in Scenario 2 above. Again, to see how accurate our predictions are, we can visualize the true vs. predicted τ (see Figure 8), seeing a relatively good fit, as well as ability to capture some of the skew.

Looking at the obtained results, we can draw a few general conclusions:

- The predictions are generally much better for the datasets that share the same initial conditions, confirming the intuition that initial conditions overpower averaging ranges in terms of their significance to turbulence modeling.
- Our model predictions are generally better the higher the Reynolds Number is. This makes intuitive sense, given that we always train on the most turbulent dataset category (Re1243-Fr20)
- The prediction is worst for Re546-Coarse8, which can be explained by the fact the lack of heat conduction actually removes one of the initial conditions (a Froude Number). The cause for these weak predictions becomes evident when looking at Figures 5 and 6. Re546 datasets show much fatter tails and their distributions barely resemble a Gaussian. Given that we are training on highly normal data with little kurtosis and skew, it makes sense that the model has trouble generalizing to data that appears to come from a different underlying distribution.

Overall, the model performs well on different datasets (based on correlation thresholds from previous research), but still shows significant variability, even when considering training on multiple initial conditions, indicating that a universal turbulence model may not be feasible. This comes somewhat expected, as it seems intuitively difficult to obtain a model that can generalize well to any change in initial conditions without ever having seen data relating to such initial conditions. Given the inherently stochastic nature of turbulence, slightly different initial conditions may not generate data that are closely related, suggesting that

it might make more sense to focus on accurate modeling with certain initial conditions.

4.2 PREDICTION OF HEAT FLUX h

Given the structural and physical similarities between τ and h_{xyz}, we refrained from performing the entire model construction and parameter testing process as we did for τ again, and fell back to using our well-performing τ assumptions to construct a model for predicting the heat flux. The final model is analogous to that presented above and is single-output, meaning it was trained separately on h_1, h_2 and h_3.

Similarly to the task above, we used the DNN to train on the different combination of datasets for h_1, h_2, h_3 and used our saved models to perform predictions on other datasets, such as all other averaging ranges within the Re1243Fr20 data category as well as on Re554Fr5, Re309Fr2 datasets. Below are the correlation values and distributions predicted by the model for the same scenarios as for τ:

1. **Scenario 1: Train single-output models on Re1243Fr20-coarse8.** Similar to above, we train our base model on a single initial condition and averaging range, and making predictions for the below datasets at time step T1. Please note that predictions within the Re546 category are not possible, given that it excludes heat conduction.

Dataset Category	h_1	h_2	h_3
Re1243Fr20-coarse16	0.7428	0.5959	0.7966
Re1243Fr20-coarse24	0.6656	0.5257	0.7800
Re1243Fr20-coarse8	0.9076	0.7913	0.8768
Re309Fr2-coarse16	0.2707	0.4785	0.6446
Re309Fr2-coarse24	0.2051	0.3976	0.6213
Re309Fr2-coarse8	0.2891	0.4797	0.6329
Re554Fr5-coarse16	0.2967	0.4901	0.6883
Re554Fr5-coarse24	0.2633	0.4080	0.6644
Re554Fr5-coarse8	0.3570	0.5032	0.6547

Resulting distributions can be seen in Figure 8. We can see that a trend is emerging, different from what we observed for τ. Namely, generalization outside of the initial condition that the model was trained on seems very difficult, especially for h_1 and h_2. An interesting insight from Figure 9 is the extreme right skew of h_3, which makes the model's ability to generalize well for that heat flux even more impressive and unexpected. Part of the difficulties for h_1 and h_2 arise from a fact similar to what we had for τ. The dataset that was trained on is highly normally distributed, with very few outliers, and essentially no kurtosis or skew. However, those statistical characteristics are not consistent across datasets (some have high skew, others have higher kurtosis), making generalization very difficult.

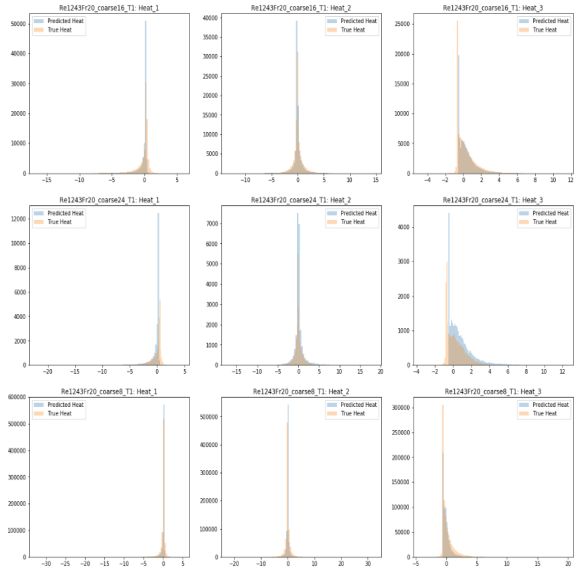


Fig. 9: Select dataset categories showing true vs. predicted h distributions for single-output model trained on Re1243Fr20-coarse8

2. **Scenario 2: Train single-output models on Re1243Fr20-coarse8 and Re309Fr2-coarse8.** Similar to the above, we trained a model across initial conditions, which should allow us to remedy some of the difficulty of the single condition model in strategy 1 to generalize well across conditions. We summarize the performance for select datasets at time step T1 below, with the distributions shown in Figure 10. We can see a significant improvement in the model's ability to generalize across initial conditions. Whereas the previous model showed a maximum correlation of around 0.35 for a dataset outside of the Re1243Fr20 category, that number has more than doubled to around 0.73. Although the improvements are not immediately evident from Figure 10, we do observe that the mean value (around 0) is predicted a lot fewer times, indicating that more of the outliers and tails of the distributions were able to be captured.

Dataset Category	h_1	h_2	h_3
Re1243Fr20-coarse16	0.7575	0.6430	0.8034
Re1243Fr20-coarse24	0.6707	0.5760	0.7886
Re1243Fr20-coarse8	0.8849	0.7688	0.8471
Re309Fr2-coarse16	0.5809	0.6134	0.8028
Re309Fr2-coarse24	0.4727	0.5293	0.7618
Re309Fr2-coarse8	0.7382	0.7492	0.8694
Re554Fr5-coarse16	0.5326	0.5947	0.7858
Re554Fr5-coarse24	0.4783	0.5265	0.7433
Re554Fr5-coarse8	0.5807	0.6226	0.7684

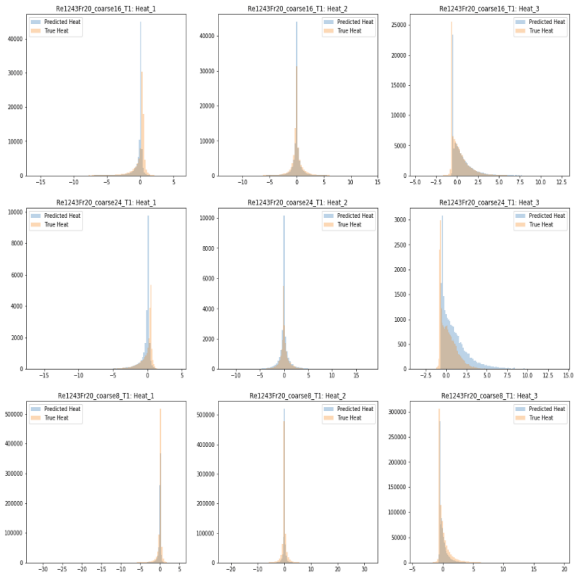


Fig. 10: Select dataset categories showing true vs. predicted h distributions for single-output model trained on Re1243Fr20-coarse8 and Re309Fr2-coarse8

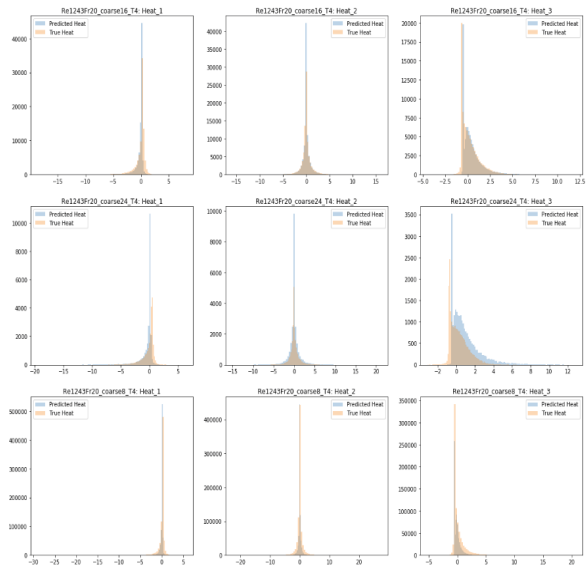


Fig. 11: Select dataset categories showing true vs. predicted h distributions for single-output model trained on Re1243Fr20-coarse8 and Re309Fr2-coarse8 at T1 and T2

3. **Scenario 3: Train single-output models on Re1243Fr20-coarse8 and Re309Fr2-coarse8 at T1 and T2.** Once again, we attempted to add a time component to our modeling without the use of recurrences, and predicting on a differing time step (T4), in order to identify how well we can generalize across the temporal dimension. Correlations are shown below, with distributions appearing in Figure 11.

Dataset Category	h_1	h_2	h_3
Re1243Fr20-coarse16	0.7570	0.6500	0.8021
Re1243Fr20-coarse24	0.6861	0.5967	0.7879
Re1243Fr20-coarse8	0.8412	0.6615	0.7988
Re309Fr2-coarse8	0.6350	0.6637	0.8045

Results appear similar to the τ case presented earlier, with the downside that the loss of accuracy for the Re309Fr2 dataset category is much more pronounced. Based on this analysis, adding data from multiple time steps does not seem favorable, as some distortion is added for certain dataset categories without observing any significant correlations increasing. However, more data at different time steps for all dataset categories would be needed to draw a definitive conclusion on whether time-dependencies should be added without using recurrences.

We can see a general pattern of predictions for h_3 being consistently accurate and significantly better across initial conditions and time, therefore indicating that the model can generalize them well to previously unseen conditions. This can indicate that h_3 shows the least variability when varying initial conditions. Issues similar to modeling τ arise, as that the model is not fully able to capture variations in skew and kurtosis. For different initial conditions, the prediction quality generally decreases significantly, although some of that effect is mitigated by training on the most and the least turbulent dataset. Once again, a universal model that approximates heat flux well across time and across differing initial conditions might not be feasible, for similar reasons as discussed for τ above, indicating that most of the effort might be better spent optimizing models within certain initial conditions.

5 LIMITATIONS & FUTURE WORK

The limitations of our analysis can be categorized into two major buckets: data-related and social. First, our preprocessing method split the data into input-height x input-width x input-depth number of boxes of size dxdxd, which will quickly cause memory issues, especially when combining datasets of differing initial conditions or smaller coarse size. We partially mitigated those issues by saving and consequently re-loading shards of data from disk, with the drawback being that it significantly slows down the training process. More computational resources in the form of higher RAM and more powerful GPU's could improve the training and validating time substantially.

Second, we had a very limited amount of data, both across space and time. Additional spatial boxes could not only improve our modeling accuracy, but would also allow us to introduce a separate convolutional network that essentially replicates our pre-processing and DNN architecture, but takes the full DNS-averaged output as input. This would not only solve the memory issue discussed above and speed up computations, but also allow us to introduce data dependencies beyond the small dxdxd boxes

that we are currently working with. Unfortunately, the current limited amount of data (one box for each of the 8 time steps) does not allow for sufficient training, as each full box would count as an input to the CNN, thereby severely limiting the number of times weight can be updated. More data would also allow us to train a more accurate recurrent network, by adding additional time steps, which would allow us to build a much more robust model across time dimensions, but identifying long-term turbulence dependencies.

Third, one of the biggest challenges of this project was reproduceability and explainability of the problem at hand and our methodology. This constitutes a big problem in scientific research in general: making people who are not necessarily the intended audience understand the relevance and importance of the work. Given our work's potential significance in issues such as climate modeling which impact people far beyond the research community, we spent a significant amount of time researching and clearly explaining the field of turbulence modeling in general, the contributions of our project to the field and why it matters. Due to our initial troubles understanding the data and its source, we further created a graphical approach (see Fig. 1) to explain where our data comes from, what it looks like and what it exactly means. The point we were trying emphasize was that one does not need to be a physicist in order to understand what we are trying to accomplish and that our analysis and its conclusions are relevant to many issues we face on a day-to-day basis.

While we have taken a big step towards identifying turbulence patterns and generalizing them across time and initial condition, future work can build on our analysis, by modeling more data, at more time-steps and using more diverse inputs. In this project, we have already experimented with training models on multiple initial conditions and averaging ranges in order to identify more general patterns of turbulence. Further work needs to be done in order to formalize those methods and test multiple conditions, with the hope of developing a universal turbulence model for sheer stress and heat flux.

6 CONCLUSION

Our project has extended prior research by attempting to model sub-grid scale turbulence characteristics in Large-Eddy Simulations, removing the need to run computationally expensive Direct Numerical Simulations. Our initial approach used a basic dense neural network with results comparable to prior research for both heat flux and sheer stress in terms of accuracy, ability to map true distributions. We subsequently introduced experimental methodologies using multi-input models to train on different initial conditions, multi-output models to speed up computations and allow for weight sharing, as well as recurrent, time-dependent models, with results showing significant potential for constructing models that generalize across time and initial DNS conditions.

7 ACKNOWLEDGMENTS

We would like to acknowledge Professor Pierre Gentine and Yu Cheng for their continued support and guidance throughout the semester, and Professor Andreas Müller for his constructive feedback and involvement.

8 REFERENCES

[1] R. A. Clark, J. H. Ferziger, and W. C. Reynolds. Evaluation of subgrid models using an accurately simulated turbulent flow. *J Fluid Mech*, 91, 04 1979.

[2] J. B. Deardorff. A numerical study of three-dimensional turbulent channel flow at large reynolds numbers. *J. Fluid Mech.*, 41, 05 1970.

[3] M. Gamahara and Y. Hattori. Searching for turbulence models by artificial neural network. *Physical Review Fluids*, 2, 06 2016.

[4] M. Giometto, A. Lozano-Duran, P. Moin, and G. I. Park. Three-dimensional transient channel flow at moderate reynolds numbers: Analysis and wall modeling. 01 2017.

[5] Q. Li, P. Gentine, J. P. Mellado, and K. A. McColl. Implications of nonlocal transport and conditionally averaged statistics on moninobukhov similarity theory and townsends attached eddy hypothesis. *Journal of the atmospheric sciences*, 75, 10 2018.

[6] C. Meneveau and J. Katz. Scale-invariance and turbulence models for large-eddy simulation. *Annual Review of Fluid Mechanics*, 32:1–32, 01 2000.

[7] P. Moin and J. Kim. Numerical investigation of turbulent channel flow. *J. Fluid Mech.*, 118, 06 1982.

[8] J. Smagorinsky. General circulation experiments with the primitive equation. i. the basic experiment. *Monthly Weather Review*, 91, 1963.