This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

SAND2019-5417C

Near-Wall Modeling using Coordinate Frame Invariant Representations and Neural Networks

Nathan E. Miller*, Matthew Barone†, Warren L. Davis IV‡, Jeffrey A. Fike§
Sandia National Laboratories¶, Albuquerque, NM, 87123

Near-wall turbulence models in Large-Eddy Simulation (LES) typically approximate near-wall behavior using a solution to the mean flow equations. This approach inevitably leads to errors when the modeled flow does not satisfy the assumptions surrounding the use of a mean flow approximation for an unsteady boundary condition. Herein, modern machine learning (ML) techniques are utilized to implement a coordinate frame invariant model of the wall shear stress that is derived specifically for complex flows for which mean near-wall models are known to fail. The model operates on a set of scalar and vector invariants based on data taken from the first LES grid point off the wall. Neural networks were trained and validated on spatially filtered direct numerical simulation (DNS) data. The trained networks were then tested on data to which they were never previously exposed and comparisons of the accuracy of the networks' predictions of wall-shear stress were made to both a standard mean wall model approach and to the true stress values taken from the DNS data. The ML approach showed considerable improvement in both the accuracy of individual shear stress predictions as well as produced a more accurate distribution of wall shear stress values than did the standard mean wall model. This result held both in regions where the standard mean approach typically performs satisfactorily as well as in regions where it is known to fail, and also in cases where the networks were trained and tested on data taken from the same flow type/region as well as when trained and tested on data from different respective flow topologies.

I. Introduction

Thanks to the proliferation of both experimental and high-quality simulation data along with the advent of modern machine-learning (ML) methods, the application of data science for the development of models in fluid dynamics is of growing interest (*e.g.*, [1]). A variety of approaches have been proposed for using ML in modeling everything from Reynolds stress components [2, 3] to unsteady pressure spectra [4] to near-wall turbulent structure [5]. The focus here is on using ML methods to develop a near-wall model for use in Large-Eddy Simulation (LES). LES near-wall turbulence modeling has received considerable attention in the literature [6] and typically

*Technical Staff, Aerosciences Department, nmille1@sandia.gov
†Technical Staff, Aerosciences Department mbarone@sandia.gov
‡Technical Staff, Scalable Analysis and Visualization Department wldavis@sandia.gov
§Technical Staff, Aerosciences Department jafike@sandia.gov
¶This paper is declared a work of the U.S. Government and is not subject to copyright protection in the United States. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

reduces to some solution of the mean flow equations near the wall, from which a wall shear stress is derived and applied as a surface boundary condition. This approach generally works well if the near-wall LES grid cell is sufficiently large so that a local Reynolds average is appropriate [7]. The assumption is invalid when the near-wall mesh spacing is not much larger than the energy-containing eddies of the near-wall turbulence being modeled. Even when this condition is met, near-wall models tend to be less accurate when the modeled flow is separated from the wall.

The goal of this work is a general wall shear stress model that behaves at least as well as a local law-of-the-wall boundary condition model in regions of simple attached boundary layers (as it would have the same input data available), but that is better suited to predict the wall shear stress vector ($\tau_{w,i}$) in regions where traditional law of the wall is known to fail. In order to be applicable for all possible flow conditions, coordinate frames, wall surface orientations, and grid types, the model must obey coordinate frame invariance and operate as a function of some set of scalar, vector, and tensor features defined at only a single point off the wall.

Invariant theory is a widely used approach in continuum mechanics for development of constitutive models. The theory can be used to derive general polynomial representations for tensor functions in terms of a general set of independent variables, of scalar, vector, and/or tensor form [8]. The number of tensors that form a so-called "integrity basis" is determined based on the particular flow features used as inputs to the model, as well as specific symmetries imposed by the system. The resulting polynomial for $\tau_{w,i}$ is called a *representation*. Based on the literature [9], the system symmetries were identified for this application and the appropriate number of form-invariant tensors were derived to build a polynomial representation (Sect. II). Note that a similar approach was identified and used by Ling *et al.* [2] to develop a machine-learned non-linear RANS model.

The invariant tensor basis formalism provides a framework for building a model that predicts the wall shear stress given a specified set of input features. It remains to choose the input features and then develop a mapping between these input features and the free parameters of the tensor representation. For this we used filtered direct numerical simulation (DNS) data to train a neural network that approximates this mapping. The DNS data were also utilized as the primary data source for testing the derived model (Sect. III). The DNS data were filtered at typical LES scales, the invariants that become the inputs to the representation were calculated from the features taken from the filtered data, and specialized neural networks were trained to use those invariants to predict the DNS-determined "true" wall shear stress vector. The specialized networks were Siamese Multi-layer Perceptrons (S-MLP), which were chosen out of a need for a specific network architecture that could take multiple sets of inputs, operate on them separately, and then combine them in the form of the tensor polynomial to achieve a vector output (Sect. IV).

The trained S-MLPs were tested on a subset of the data which was not used in the training process, and the wall shear stress predictions were compared to the predictions that would be made by a traditional law-of-the-wall boundary condition model given the same filtered state variables (Sect. V).

II. Invariant Representation

The target output for the new near-wall model is the wall shear stress vector ($\tau_{w,i}$), and thus the chosen representation is a vector-valued polynomial function of input vectors and tensors, yet to be specified. The geometry of the problem reveals that isotropy of the resulting functional form need not be preserved in all directions, since the presence of the wall provides a "preferred direction,"

that of the local wall unit normal vector. This is analogous to the situation of "transverse isotropy" in solid mechanics, which is used to model, for example, constitutive relations for materials that have a single preferred direction, such as unidirectional fiber composites [10, 11]. The particular form of transverse isotropy used here is termed *rotational symmetry*, which results in a model that is invariant to coordinate frame rotations about the preferred direction.

The equation for wall shear stress is written as a linear combination of a set of vector-valued, transversely-isotropic functions of a set of input features:

$$\tau_{w,i} = G^{(1)}\Pi_i^{(1)} + G^{(2)}\Pi_i^{(2)} + G^{(3)}\Pi_i^{(3)} + ... + G^{(n)}\Pi_i^{(n)}. \tag{1}$$

Here, the vector-valued functions $\Pi_i^{(k)}$ are called the form invariants, and these are weighted by scalar coefficients $G^{(k)}$. The form invariants are functions of the chosen input features, defined subsequently. The coefficients, $G^{(k)}$, are functions of scalar invariants, λ_j, derived from those same input features:

$$G^{(k)} = \mathcal{F}^{(k)}(\lambda_1, \lambda_2, \lambda_3, ..., \lambda_m). \tag{2}$$

The input features were chosen as the filtered velocity vector $\overline{\mathbf{U}}$, the local filtered rate of strain tensor ($\overline{\mathbf{S}}$), and the local filtered rotation rate tensor ($\overline{\mathbf{\Omega}}$), at a near-wall grid point (or wall layer interface point) along the direction of the wall-normal vector ($\mathbf{n}$) associated with the location where $\tau_{w,i}$ is evaluated.

Since the input features for the defined representation include one symmetric tensor ($\overline{\mathbf{S}}$), one skew-symmetric tensor ($\overline{\mathbf{\Omega}}$), and two vectors ($\overline{\mathbf{U}}$ and $\mathbf{n}$), a total of 20 scalar invariants λ are defined in the literature for rotational symmetry about $\mathbf{n}$ [10, 11, 12]:

$$\begin{aligned}
\lambda_1 &= \left\{\overline{\mathbf{S}}\right\}, & \lambda_2 &= \left\{\overline{\mathbf{S}}^2\right\}, \\
\lambda_3 &= \left\{\overline{\mathbf{\Omega}}^2\right\}, & \lambda_4 &= \left\{\overline{\mathbf{S}}^3\right\}, \\
\lambda_5 &= \left\{\overline{\mathbf{S}}\,\overline{\mathbf{\Omega}}^2\right\}, & \lambda_6 &= \overline{\mathbf{U}} \cdot \overline{\mathbf{U}}, \\
\lambda_7 &= \overline{\mathbf{U}} \cdot \overline{\mathbf{S}}\,\overline{\mathbf{U}}, & \lambda_8 &= \overline{\mathbf{U}} \cdot \boldsymbol{\epsilon}\overline{\mathbf{\Omega}}, \\
\lambda_9 &= \mathbf{n} \cdot \overline{\mathbf{S}}\mathbf{n}, & \lambda_{10} &= \mathbf{n} \cdot \overline{\mathbf{U}}, \\
\lambda_{11} &= \mathbf{n} \cdot \boldsymbol{\epsilon}\overline{\mathbf{\Omega}}, & \lambda_{12} &= \mathbf{n} \cdot \overline{\mathbf{S}}^2\mathbf{n}, \\
\lambda_{13} &= \mathbf{n} \cdot \boldsymbol{\epsilon}\overline{\mathbf{S}}\,\overline{\mathbf{\Omega}}, & \lambda_{14} &= \mathbf{n} \cdot (\overline{\mathbf{S}}\mathbf{n} \times \overline{\mathbf{S}}^2\mathbf{n}), \\
\lambda_{15} &= \mathbf{n} \cdot \boldsymbol{\epsilon}\overline{\mathbf{S}}\,\overline{\mathbf{\Omega}}^2, & \lambda_{16} &= \mathbf{n} \cdot \overline{\mathbf{S}}\,\overline{\mathbf{\Omega}}\mathbf{n}, \\
\lambda_{17} &= \mathbf{n} \cdot \overline{\mathbf{S}}\,\overline{\mathbf{U}}, & \lambda_{18} &= \mathbf{n} \cdot (\overline{\mathbf{U}} \times \overline{\mathbf{S}}\,\overline{\mathbf{U}}), \\
\lambda_{19} &= \mathbf{n} \cdot (\overline{\mathbf{U}} \times \overline{\mathbf{S}}\mathbf{n}), & \lambda_{20} &= \mathbf{n} \cdot \overline{\mathbf{\Omega}}\,\overline{\mathbf{U}},
\end{aligned} \tag{3}$$

where the $\{\}$ signifies the trace and $\boldsymbol{\epsilon}$ is the permutation tensor[a].

The individual form-invariants that constitute the basis can be derived from the list of defined scalar invariants via a lengthy process that can be found in Spencer [13]. That process is not included here because the form-invariants are tabulated in the literature [10, 11, 12]. A total of

[a]Also known at the Levi-Civita tensor, ϵ_{ijk} is unity for cyclic permutations of (1,2,3) for the indices and is -1 for anticyclic permutations, *e.g.*, ϵ_{231}=1 while ϵ_{213}=-1.

eight Π's are defined as,

$$\begin{aligned} \Pi^{(1)} &= \overline{\mathbf{U}}, & \Pi^{(2)} &= \boldsymbol{\epsilon}\overline{\boldsymbol{\Omega}}, \\ \Pi^{(3)} &= \mathbf{n}, & \Pi^{(4)} &= \overline{\mathbf{S}}\mathbf{n}, \\ \Pi^{(5)} &= \mathbf{n} \times \overline{\mathbf{S}}\mathbf{n}, & \Pi^{(6)} &= \overline{\boldsymbol{\Omega}}\mathbf{n}, \\ \Pi^{(7)} &= \mathbf{n} \times \overline{\mathbf{U}}, & \Pi^{(8)} &= \mathbf{n} \times \overline{\boldsymbol{\Omega}}\mathbf{n}. \end{aligned} \tag{4}$$

Zheng [12] does not include $\Pi^{(8)}$ in its list, but it is included in Zheng [11]. It may be that including $\Pi^{(8)}$ means that this list of form-invariants is not a minimal basis, but it is not necessarily required that this list be irreducible and therefore $\Pi^{(8)}$ was included in the analysis.

When put together, the representation takes the form,

$$\begin{bmatrix} \tau_{w,1} \\ \tau_{w,2} \\ \tau_{w,3} \end{bmatrix} = G^{(1)} \begin{bmatrix} \Pi_1^{(1)} \\ \Pi_2^{(1)} \\ \Pi_3^{(1)} \end{bmatrix} + G^{(2)} \begin{bmatrix} \Pi_1^{(2)} \\ \Pi_2^{(2)} \\ \Pi_3^{(2)} \end{bmatrix} + G^{(3)} \begin{bmatrix} \Pi_1^{(3)} \\ \Pi_2^{(3)} \\ \Pi_3^{(3)} \end{bmatrix} + \cdots + G^{(8)} \begin{bmatrix} \Pi_1^{(8)} \\ \Pi_2^{(8)} \\ \Pi_3^{(8)} \end{bmatrix}, \tag{5}$$

or, as a matrix operation,

$$\begin{bmatrix} \tau_{w,1} \\ \tau_{w,2} \\ \tau_{w,3} \end{bmatrix} = \begin{bmatrix} \Pi_1^{(1)} & \Pi_1^{(2)} & \Pi_1^{(3)} & \cdots & \Pi_1^{(8)} \\ \Pi_2^{(1)} & \Pi_2^{(2)} & \Pi_2^{(3)} & \cdots & \Pi_2^{(8)} \\ \Pi_3^{(1)} & \Pi_3^{(2)} & \Pi_3^{(3)} & \cdots & \Pi_3^{(8)} \end{bmatrix} \begin{bmatrix} G^{(1)} \\ G^{(2)} \\ G^{(3)} \\ \vdots \\ G^{(8)} \end{bmatrix}. \tag{6}$$

The scalar coefficients are unknown and the functions ($\mathcal{F}^{(k)}$) that relate them to the scalar invariants are still to be determined. A machine learning technique, specifically the Siamese Multi-layer Perceptron (S-MLP) described in Sect. IV, is used to approximate these functions.

Before calculating the scalar invariants and form-invariants to be used in the representation, the input features were nondimensionalized to ensure some generality to the model application and to prevent the illogical combination of units within the functions. All quantities were normalized using outer variables, specifically, the freestream velocity (U_∞), the freestream flow density (ρ_∞), and a problem-specific characteristic geometric length scale (h). From here on, all uses of $\overline{\mathbf{U}}$, $\mathbf{n}$, $\overline{\mathbf{S}}$, and $\overline{\boldsymbol{\Omega}}$ have been nondimensionalized with those outer variables. We note that choice of non-dimensionalization may limit or otherwise impact the accuracy of the models for flows that are very different from the training data, for example, training on low-Reynolds number flow and predicting at a higher Reynolds number. We do not examine these scaling issues in the present work.

III. DNS Data, Filtering, and Pre-processing

Pseudo-LES data were produced from data from two different DNS datasets generated using Sandia National Laboratories' structured, multi-block, finite volume code, SIGMA-CFD. Details on SIGMA-CFD can be found in multiple previous publications including Arunajatesan *et al.* [14] and Barone and Arunajatesan [15]. Two different flow cases—a backward facing step and a cavity flow—were simulated via DNS. LES-style filtering was then applied to the DNS data to produce pseudo-LES data for training and testing the ML models.

A. DNS Details

The first data set considered was the DNS of the flow over a backward facing step at a Mach number of 0.6. The DNS domain in the region upstream of the step was $97\delta_0 \times 1200\delta_0 \times 6\delta_0$ in the streamwise $\times$ wall-normal $\times$ spanwise directions, respectively, where δ_0 is the boundary layer thickness for a precursor RANS simulation from which a solution profile was extracted and used for the inlet condition for the DNS. The selection of the length of $97\delta_0$ for the domain extent upstream of the step was based on a separate Mach 0.6 boundary layer simulation, and was the distance at which the local boundary layer thickness reached a nominal value of $\delta = 3\delta_0$. The boundary layer was made turbulent well upstream of the step by using a synthetic turbulence method that introduces disturbances within the boundary layer to excite transition to turbulence. In the backward facing step simulation the boundary layer thickness was also $\approx 3\delta_0$ at the step edge, though slightly smaller due to the favorable streamwise pressure gradient produced by the presence of the step. The wall-normal extent of the domain was tested to ensure that the domain boundary was not adversely affecting the boundary layer growth, and although very little differences were observed in the boundary layer with wall-normal domain dimensions ranging from $60\delta_0$ to $1200\delta_0$, data from the largest tested domain are used here. The step height was chosen to be $h = 3\delta_0$, resulting in $h/\delta \approx 1$ at the step. The Reynolds number, based on the velocity at the edge of the boundary layer and h, was 6254 at the step edge. Downstream of the step, the domain extended $130\delta_0$ in the streamwise direction, which was well beyond the flow's reattachment point, located at about $20\delta_0$ downstream of the step. A total of ≈ 237 million grid cells were used in the domain with horizontal and vertical grid stretching allowing for high concentrations of cells near the walls, in the shear layer, and near the reattachment point. The backward facing step was first simulated for 6.5 flow-through times with a time step of 1×10^{-9} s, followed by an additional 6 flow-through times during which data were saved off to be used for training and testing the NN models.

Cavity flow was simulated using an upstream mesh and dimensions with the same streamwise and spanwise dimensions as that used for the backward facing step simulation ($97\delta_0 \times 6\delta_0$) as well as the same upstream Mach 0.6 boundary layer. The vertical extent of the domain was reduced to $120\delta_0$ with a sponge layer starting at $60\delta_0$ to damp out pressure waves produced at the downstream cavity lip. The upstream boundary layer grew, without the presence of a favorable pressure gradient, to a height of $3\delta_0$ at the cavity edge. The cavity was designed with a depth of $D = 3\delta_0$, identical to the step, and thus the Reynolds number at the cavity lip was identical to the number at the step edge. The cavity had a streamwise length of $L = 9\delta_0$, thus producing a cavity aspect ratio of $L/D = 3$. Downstream of the cavity, the domain extended $109\delta_0$. A total of ≈ 134 million grid cells were used and again vertical and horizontal grid stretching were used to cluster cells near the walls, in the shear layer, and in the cavity. The cavity simulation also used a time step of 1×10^{-9} s and was run for 12 flow-through times to remove transients prior to an additional 4.5 flow-through times during which data were saved for later use.

B. Pseudo-LES Filtering

The DNS data were filtered down to a resolution that would be similar to that used for a wall-modeled LES of the same flow case. In this way pseudo-LES data were produced that could be used as the input data for the near-wall model, while the filtered DNS wall shear stress vector became the labels to be predicted by the model.

For the backward facing step DNS data, the domain was broken into two regions which were both treated independently for training and testing individual S-MLP ensembles as well as being used in a single, comprehensive data set to train and test a third network. The first region was from

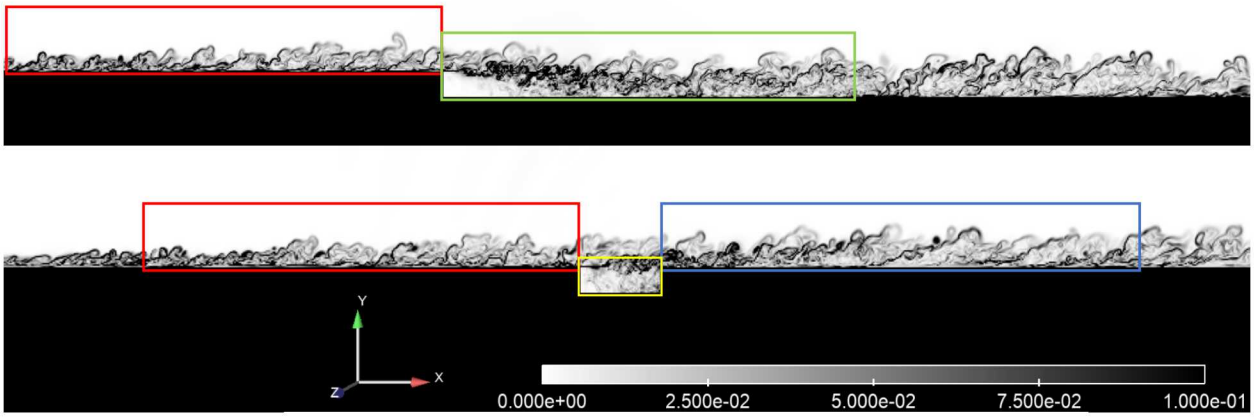

Figure 1: Density gradient contours from the DNS simulations of the backward facing step (top) and the cavity flow cases (bottom). The upstream and recirculation regions of the backward facing step flow are bounded by the red and green boxes, respectively, while the upstream, cavity, and downstream regions of the cavity flow case are bounded by the red, yellow, and blue boxes, respectively.

well upstream of the step ($\approx 54\delta_0$ upstream) to the step edge, while the second was the detached recirculation region defined along the step face and along the wall below the step downstream to $\approx 47\delta_0$ downstream of the step (Fig. 1). For the cavity flow DNS data, three separate domains were defined as the upstream region (similar to that defined for the backward facing step), the downstream region (extending downstream from the downstream lip of the cavity), and the cavity itself, composed of both the upstream and downstream cavity wall faces as well as the cavity floor.

In the region upstream of the backward facing step two sets of filter widths were used. The filter halfwidths of these sets corresponded to 10×4 and 20×8 DNS grid points in the x (streamwise) and z (spanwise) directions, respectively. This corresponded to nominal pseudo-LES control volume sizes of $[2\Delta^+_{f,x}, 2\Delta^+_{f,z}] \approx [80, 50]$ and $[2\Delta^+_{f,x}, 2\Delta^+_{f,z}] \approx [170, 110]$, respectively, where $\Delta^+_{f,i}$ is the inner-scaled filter halfwidth in the ith direction and the friction velocity was taken as the mean over the upstream region. These scales are within the nominal range of scales recommended by Larsson *et al.* [6] for wall-modeled LES. Pseudo-LES control volume centers were defined in the streamwise and spanwise directions based on multiples of the respective filter widths resulting in a grid of 62×19 and 30×9 points in the x and z directions, respectively, for the 10×4 and 20×8 filter sizes, respectively. A simple box filter operation was applied, consistent with the common approach in finite-volume based LES simulations where the cell control volume implies the filtering operator. Four different nominal wall-normal distances to the center of the first wall-normal cell were used and the filtering process was applied from the wall to twice that nominal distance. The four nominal heights were $y^+ = [30, 60, 100, 150]$. For each control volume defined by each filter width set and each nominal wall-normal cell center distance, the filtered velocity components and the gradients of the filtered velocity were calculated and then used to compute the scalar and form-invariants given in equations (3) and (4). Collectively this resulted in nearly 5800 usable pseudo-LES grid cells that could be used in the upstream region of each individual DNS frame. A total of 50 independent DNS frames for the backward facing step flow were used for this work. The frames were separated in time by a minimum of $0.51h/U_\infty$ although most were separated by $2.55h/U_\infty$.

In the detached, recirculation region behind the step, only a single filter set was used in the wall-parallel directions along with three wall-normal cell center distances. The single filter width set was defined with halfwidths of 4×4 DNS grid points and was applied both on the step face and along the "floor" below the step. This led to 300 usable control volumes in the pseudo-LES grid along the step face (15×20 in the $y \times z$ directions, respectively) and 2980 usable volumes in the grid defined along the floor (149×20 in the $x \times z$ directions, respectively). The three wall-normal cell center heights were defined based on the idea that an LES grid for this flow would likely be defined based on choosing a desired wall-normal resolution in the region downstream of the point at which the boundary layer reattached to the wall. Three heights of nominal y^+ values of $[15, 30, 60]$ were chosen and the appropriate physical wall-normal distances were defined and propagated back upstream towards the step, consistent with the notion of a Cartesian LES mesh. This resulted in the cell-centers near the step having the same physical distance to the wall as in the reattached region, but due to the comparably low wall shear stress values behind the step, they have much lower actual y^+ values. A traditional law-of-the-wall-based wall model might incorrectly assume these low y^+ values justify a linear viscous-sublayer extrapolation for determining the wall shear stress instead of using some other law appropriate for detached flows. The same three physical wall-normal distances were used on both the "floor" below the step and on the step face. Collectively this resulted in 9840 pseudo-LES grid cells within the recirculation region for each of the 50 DNS frames.

The upstream and downstream regions for the cavity case were treated similarly to the upstream region of the backward facing step case (Fig. 1). The same two sets of filter sizes were used for the wall-parallel directions (10×4 and 20×8 DNS grid points) along with three nominal heights of $y^+ = [30, 60, 100]$. These filter set sizes resulted in pseudo-LES grids of $62 \times 19 \times 3$ and $30 \times 9 \times 3$ points in the upstream region—which had the same streamwise dimension, $54\delta_0$, as was used in the backward facing step case—for the 10×4 and 20×8 grid-point filters, respectively. The downstream region had a streamwise dimension of $64\delta_0$ resulting in pseudo-LES grids of $74 \times 19 \times 3$ and $36 \times 9 \times 3$ points for the two grid-point filter sets, respectively. Collectively this resulted in over 4300 pseudo-LES grid cells in the upstream region and 5190 in the downstream region of the cavity flow case. A total of 70 independent DNS frames, all separated in time by $2.55h/U_\infty$, were used from the cavity flow simulation.

In the actual cavity itself, a single filter set of 4×4 DNS grid points was used on both cavity walls and on the cavity floor. Three wall-normal heights were used and were defined as equal to the three heights used in the recirculation region of the backward facing step flow, which were defined based on nominal y^+ values defined in the reattached region. This resulted in a usable grid of $32 \times 21 \times 3$ pseudo-LES grid cells on the cavity floor with $14 \times 21 \times 3$ grid cells on each cavity wall, thus resulting in 3780 total grid points available from each of the 70 DNS frames.

C. Data Scaling and Sampling Procedures

After the data were filtered at every pseudo-LES grid point and the invariants were calculated, not only was everything normalized with outer variables (U_∞, ρ_∞, and h), but additional pre-processing of the data and datasets was performed to provide better performance of the neural networks. First, the maximum vector norm of each of the Π's and for τ_w was determined from all of the available data and each respective variable was divided by the maximum norm. This resulted in the magnitude of each Π in each input vector lying within the range of zero to one. The same was true for τ_w. Additionally, the values of the individual λ's often covered multiple orders of magnitude across all the available input data. Even when these data were linearly scaled to a

negative one to one range to improve network performance, large magnitude outliers caused the distributions of the λ's to be very narrow, which is not ideal for the networks to parse. In order to provide better distributed data for the networks, all of the lambda data was transformed as,

$$\lambda_i = \text{sign}\left(\lambda_i\right) \log_{10}\left(\left|\lambda_i\right|\right), \tag{7}$$

prior to being mapped to a range of negative one to one. This provided much better distributions of the λ values within the range and improved network results.

After being re-scaled in this manner, the data were then split into groups of 80% and 20% within each of the aforementioned regions of the different flows. The 80% files were used for training and validation while the 20% were held aside for use in the final testing shown below. Joint probability density functions (PDFs) of the two non-zero components of τ_w taken from each region of the flow showed an expected behavior that the wall shear stress values near the mean are much more probable than more extreme values. In the attached boundary layer region, the PDF resembles a log-normal shape for the streamwise component of the stress and a Gaussian shape for the spanwise component, consistent with experimental measurements [16]. Prior experience showed that when neural networks were trained on data without accounting for this, the networks returned data with a normal distribution with a smaller standard deviation than that seen in the training data. Essentially, networks have a tendency to only learn the most common occurrences of τ_w (which was typically near the mean) and would predict a wall shear stress near that value regardless of the input features. The networks were not learning the low-probability events. So a strategy was developed and applied to the files containing the 80% of the data to be used for training such that feature vectors with more common wall shear stress vectors were undersampled and feature vectors with low-probability wall shear stresses were sampled repeatedly (up to 10 times) so that their representation within the training dataset was increased. Applying this strategy to each of the 80% data files for each of the regions of each flow also allowed for the total amount of training data from each region to be independently adjusted to make sure that training was not too extremely biased towards the regions or filter set sizes from which more pseudo-LES grid points were available. While the total amount of training data across all regions was kept relatively constant, the relative contribution from the datasets with the larger filter sizes was increased while the relative contribution from the datasets with the smaller filter sizes (where more data were available) was decreased. The data files with the 20% of the data that was partitioned off were not manipulated in any way so that final testing could be representative of the actual data obtained from LES simulations.

IV. Neural Network Methods

The invariant approach used here necessitated a machine-learning network architecture that could correctly utilize the scalar- and form-invariants and combine them in the correct way following Eqn. (5) such that the unknown scalar coefficients could be determined. For this, a network architecture based on the Siamese Multi-layer Perceptron (S-MLP) was used [17]. A multi-layer perceptron is simply a network of individual perceptrons (neurons) organized into an input layer, hidden layers, and an output layer. The S-MLP is composed of two MLPs organized in parallel, each with its own input and output layers and potentially its own hidden layers. The output layers of the two parallel MLPs are combined following some function and a final output is formed. For the purposes of training, an architecture must be designed that can propagate errors backwards through the combination of the individual output layers into the two parallel MLPs correctly based

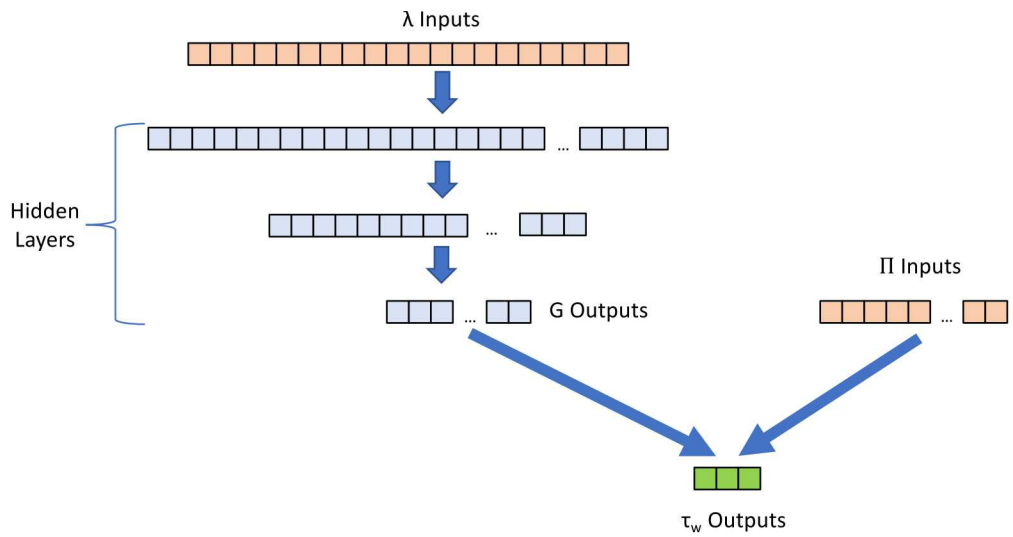

Figure 2: Diagram of the Siamese Multi-layer Perceptron architecture as designed and implemented for the invariant approach.

on the single set of outputs from the final output layer. For example, if the outputs of the first and second MLP of a S-MLP are to be summed into a final output, the architecture must pass errors backwards through that sum to appropriately correct the weights in each of the first and second MLPs in order to correct the final single output.

Our S-MLP was designed such that one MLP took all of the scalar-invariants (20 λ's) as inputs, and operated on them to produce an output layer of scalar coefficients (8 G's). The other MLP took as inputs the form-invariants (8 Π's of three components each) but did not perform any operations on them and simply passed them directly to the output layer where they could be combined with the G's following Eqn. (6) (Fig. 2). This produced the final wall shear stress output vector which could be checked against the known target. This custom architecture was then used to adjust the weights used on the one side of the S-MLP that is predicting G's, thus training the network towards minimizing the total error in predicting the target shear stresses.

The S-MLP networks were built using a combination of open source and Sandia National Laboratories' proprietary software developed for ML research. This included constructing our neural networks using, as a foundation, a combination of TensorFlow [18] and Keras[b]. TensorFlow is an open source library for computation, originally developed by Google, which focuses on tensor computations which form the core calculations in neural networks. Keras is an open source library built as an abstraction to TensorFlow, Microsoft Cognitive Toolkit (CNTK) [19], and Theano [20]. It eases construction of neural networks with the aforementioned underlying packages. Additionally, a Sandia National Laboratories' library called TensorNet, that allowed us to construct new neural networks quickly and easily from JavaScript Object Notation (JSON) definitions, was also used. It provides for flexible use of training, validation, and testing datasets, as well as cross-fold validation, convolution, and auto-encoders. Finally, we used a Sandia python library named GrouPy, which contains base data structures that facilitate data introspection and processing, such as dataset subsampling, joining, and splitting.

The actual networks used here were built with four hidden layers on the one side of the S-

[b] https://keras.io

MLP with those layers having 120, 92, 64, and 36 nodes, respectively. The final layer of that side had eight nodes which were the G values. For each of the individual flow cases and for each region within each of the flow cases, an ensemble of ten S-MLPs was trained using the uniformly sampled data taken from the 80% of the data partitioned off for training and validation. This meant that seven different ensemble networks were trained: one using all of the data from both regions of the backward facing step case, one each using data from each of the two regions of that case, one using all of the data from all three regions of the cavity flow case, and one each using data from each of the three regions of that case. Ensembles were used because each individual network was initialized with random weights and thus each converged to a slightly different solution. Using the average prediction over the ensemble ensured that a stable and more accurate result was achieved. The seven trained ensemble networks were then each tested against the respective data files containing the 20% of the data that corresponded to the data each was trained on. So the network trained with 80% of the data taken from the upstream region of the cavity flow case was then tested against the remaining 20% of the data from that region.

Additionally, a final test was performed where the network that was trained on data from all three regions of the cavity flow case was used to predict the 20% of the data taken from both regions of the backward facing step case. This was done to test the potential portability of a network trained on a range of flow topologies when applied to make predictions on a new configuration not contained within the training set but having similar topological features. Obviously a large set of other tests could be done with networks trained on one set of data and tested on another, but we restricted ourselves to only these eight tests for this work.

V. Results

The ensembles of S-MLPs generated predictions for the non-zero components of the wall shear stress ($\tau_{SMLP,i}$)[c] which were then processed to obtain the wall shear stress magnitude ($|\tau_{SMLP}|$). These components of the wall shear stress vector, and the wall shear stress magnitude, were then compared to both the true values taken from the DNS (τ_{DNS}) and to wall shear stress values that would have been predicted by a traditional LES law-of-the-wall boundary layer model, $\tau_{walllaw}$. The wall shear stress magnitudes as defined by the law of the wall ($|\tau_{walllaw}|$) were determined for each point based on the filtered velocity magnitude at the cell center using the formulation of Spalding [21]. The magnitudes were then decomposed into individual components ($\tau_{walllaw,i}$) by assuming alignment of the shear stress vector with the filtered velocity vector. Thus, the S-MLP predictions for both the components and the magnitude could be compared to both the ideal and to the law of the wall. The expectation was that these two comparative datasets constituted the "best achievable" and "worst acceptable" limits for the new model.

The quality of the different predictions was quantified via two direct metrics, the one-dimensional (1D) correlation coefficient (r) and the mean squared error (MSE). The correlation coefficient was defined as,

$$r = \frac{\sum_{i=1}^{n}(p_i - \langle p \rangle)(d_i - \langle d \rangle)}{\sqrt{\sum_{i=1}^{n}(p_i - \langle p \rangle)^2}\sqrt{\sum_{i=1}^{n}(d_i - \langle d \rangle)^2}}, \tag{8}$$

where p_i and d_i are the predictions and true DNS data, respectively, and the $\langle\ \rangle$ are means taken over all available data pairs. The MSE was defined as,

$$\mathrm{MSE} = \left\langle (p_i - d_i)^2 \right\rangle, \tag{9}$$

[c] the stress vector was assumed to be tangential to the wall for all the filtered DNS data.

Table 1: Quality metrics comparing wall shear stress component and magnitude ($|\tau_w|$) predictions from the S-MLP (τ_{SMLP}) and from the law of the wall ($\tau_{walllaw}$) to the true DNS values for data taken collectively from both regions of the backward facing step flow.

	r	MSE	K-L Div.
$\lvert\tau_{SMLP}\rvert$	0.862	5.36e-7	0.052
$\lvert\tau_{walllaw}\rvert$	0.782	1.11e-6	1.75
$\tau_{SMLP,x}$	0.924	5.21e-7	0.061
$\tau_{walllaw,x}$	0.875	9.05e-7	2.11
$\tau_{SMLP,y}$	0.714	2.37e-9	0.002
$\tau_{walllaw,y}$	0.528	2.31e-9	0.258
$\tau_{SMLP,z}$	0.674	2.79e-7	0.010
$\tau_{walllaw,z}$	0.394	4.23e-7	2.29

and is a global norm of the errors of the individual predictions. The MSE is somewhat biased towards errors in large magnitude values, *i.e.*, a smaller relative error in a large magnitude stress will have a larger impact on MSE than a larger relative error on a stress near zero. The MSE was used as the loss function in the neural network training process as well, meaning that the weights of the neural network were optimized so as to minimize the MSE.

Additionally, discrete PDFs of the data from the two different prediction techniques were also produced and the accuracy of their relative fits to the PDF of the true wall shear stress data was quantified via the Kullback-Leibler Divergence, or K-L Divergence (KLD). The K-L divergence is defined as,

$$KLD(D||P) = \sum_{k=1}^{bins} D_k \ln \frac{D_k}{P_k}, \tag{10}$$

and measures the amount of information that is lost when approximating the distribution D (the true DNS data) using the predicted distribution P. For a perfect prediction of D by P, $KLD = 0$.

A. Backward Facing Step Flow Case

The first S-MLP ensemble was trained on data from both the upstream and recirculation regions of the backward facing step flow and was tested against data taken from both regions. That network showed considerable improvement in its predictions over the law of the wall, both in the quality metrics (Table 1) and the PDFs (Fig. 3). The correlation coefficients were improved and the MSE values were reduced for the stress magnitude and for two of the three stress components: the x-component (which was always zero on the face of the step) and the z-component.

Only the MSE for the y-component of the wall shear stress did not show an improvement over the law of the wall. These values are only non-zero on the face of the backward facing step and typically have much lower magnitudes than the other components. The use of the MSE as a loss function for the neural networks likely led to the lower accuracy for these low-magnitude values. It may be possible to improve the predictions of the y-component by appropriately weighting of the loss function to remove this bias, but at the expense of higher error in prediction of the other components. The S-MLP does show an ability to predict much higher wall shear stress values than seems to be capable from the law of the wall and is therefore able to much more accurately fill

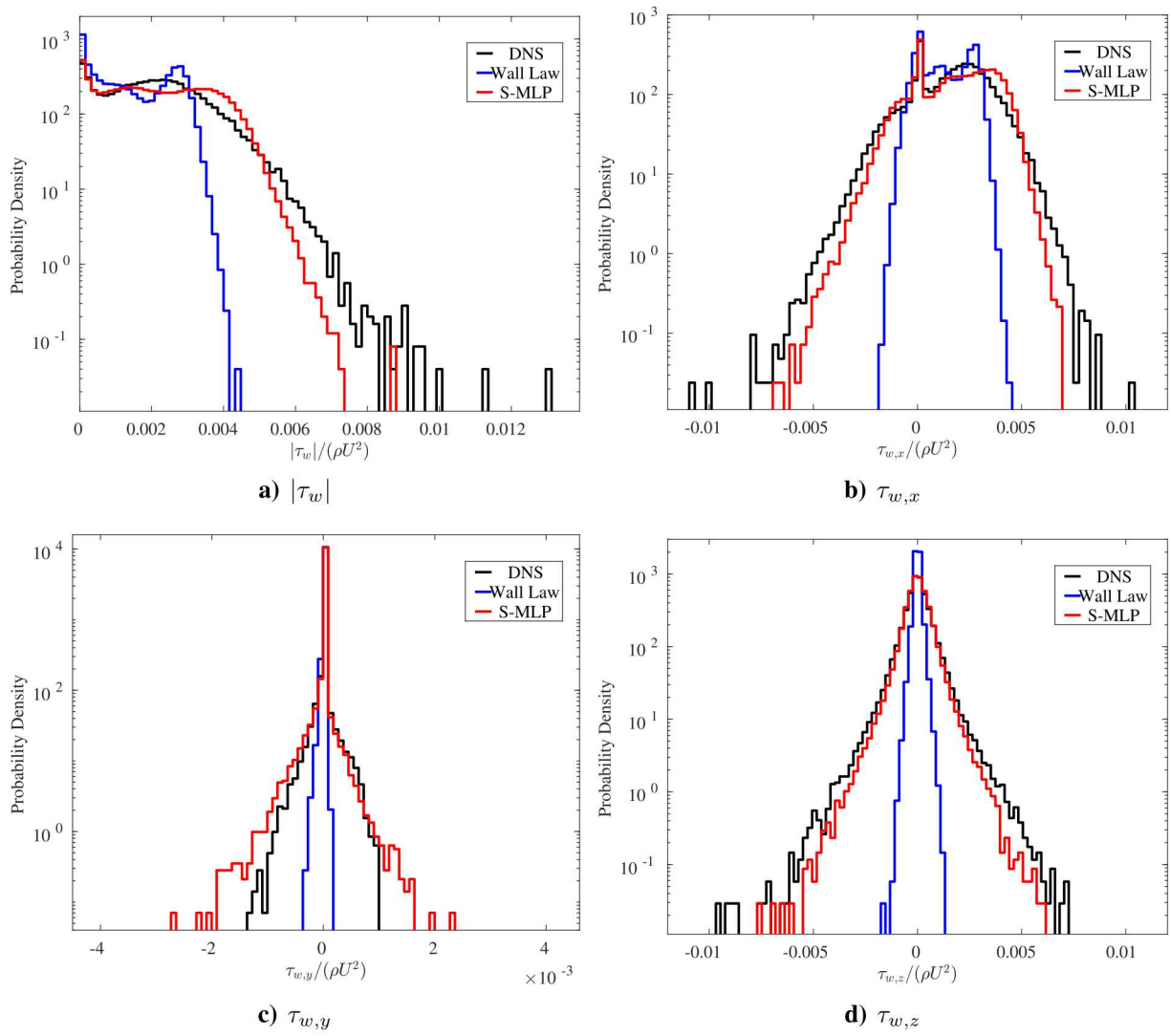

Figure 3: Probability density functions of the wall shear stress magnitude ($|\tau_w|$) and its components as predicted by the law of the wall and the S-MLP compared to the true data from filtered DNS for the backward facing step case.

out the PDFs for the magnitude and components. For $\tau_{w,y}$, although the shape of the PDF was improved (lower K-L Divergence) by using the proposed representation with the S-MLP, the MSE was slightly worse than with the law of the wall approach. This suggests that although the S-MLP does a good job estimating more extreme values of $\tau_{w,y}$ and perhaps even produces extreme values when appropriate (still has a decent r), the actual accuracy of any given extreme value is not always particularly good. So although the law-of-the-wall predictions are almost always near zero, the error is very consistent and low, where the S-MLP has larger individual errors but a better aggregate PDF. On the whole, in an LES simulation of a recirculation region, this may still be a large improvement in aggregate and may improve the performance and estimation of total forces acting on a surface of interest.

The PDF of the wall shear stress magnitude for the S-MLP shows a local peak near $|\tau_w|/(\rho U^2) \approx 0.004$ (in which it is more likely to predict that magnitude than is correct based on the DNS). This

Table 2: Same as Table 1 but for the upstream region of the backward facing step case.

	r	MSE	K-L Div.
$\lvert\tau_{SMLP}\rvert$	0.537	8.45e-7	0.194
$\lvert\tau_{walllaw}\rvert$	0.385	1.06e-6	3.77
$\tau_{SMLP,x}$	0.534	8.48e-7	0.200
$\tau_{walllaw,x}$	0.380	1.02e-6	3.75
$\tau_{SMLP,z}$	0.468	2.32e-7	0.020
$\tau_{walllaw,z}$	0.107	2.22e-7	6.88

peak is primarily attributed to a similar feature in the PDF of $\tau_{w,x}$ in the upstream region of the flow, (as will be seen again for the upstream region-only model, Fig. 4(b)).

1. *Upstream Region*

Again the S-MLP predictions for the upstream region of the backward facing step flow showed improvements over the predictions based on the law of the wall in each of the metrics (Table 2). The correlation coefficient improved, but stayed at ≈ 0.5 for the magnitude and the components of the wall shear stress. Combined with the improvements in the MSE, this suggests that although there was a general improvement in the accuracy of predicting individual stress values, the high level of r seen for the combined data set was mostly due to improvements made in the recirculation region of the flow.

The improvements in KLD were perhaps the most impressive, and especially so for the spanwise component of the wall shear stress, $\tau_{w,z}$. The PDFs for the shear stress magnitude show the considerable improvement of the S-MLP predictions over the law of the wall and are nearly indistinguishable from the PDF for the x-component of the shear stress (Fig. 4). This was not unexpected given that in the upstream region the flow was essentially that of an attached unidirectional boundary layer wherein the streamwise component should dominate the wall shear stress vector.

The S-MLP does exhibit an incorrect peak location near $\tau_{w,x}/(\rho U^2) \approx 0.004$. This is also observed in the PDF of $|\tau_w|$ for the upstream region, and corresponds to the same incorrect peak in the data based on training using the entire backward facing step case at once (Fig. 3(a) and (b)).

The PDFs for the z-component show that the law of the wall model provides a very narrow distribution for $\tau_{w,z}$, while the S-MLP predicts an almost perfect distribution of $\tau_{w,z}$ (Fig. 4(c)). Interestingly, although the PDF of the S-MLP for $\tau_{w,z}$ is nearly perfect, the correlation coefficient is still relatively unimpressive, thus suggesting that the S-MLP still misses a considerable number of individual predictions, but does so in a way that produces an appropriate statistical spread of predictions that leads to an accurate PDF.

2. *Recirculation Region*

In the recirculation region, the S-MLP again outperformed the law of the wall approach in predicting the wall shear stress magnitude and all three components of the wall shear stress vector (Table 3). The fact that the network trained on data from only the recirculation region is better at predicting $\tau_{w,y}$ than was the law of the wall while the network that was trained on data from everywhere

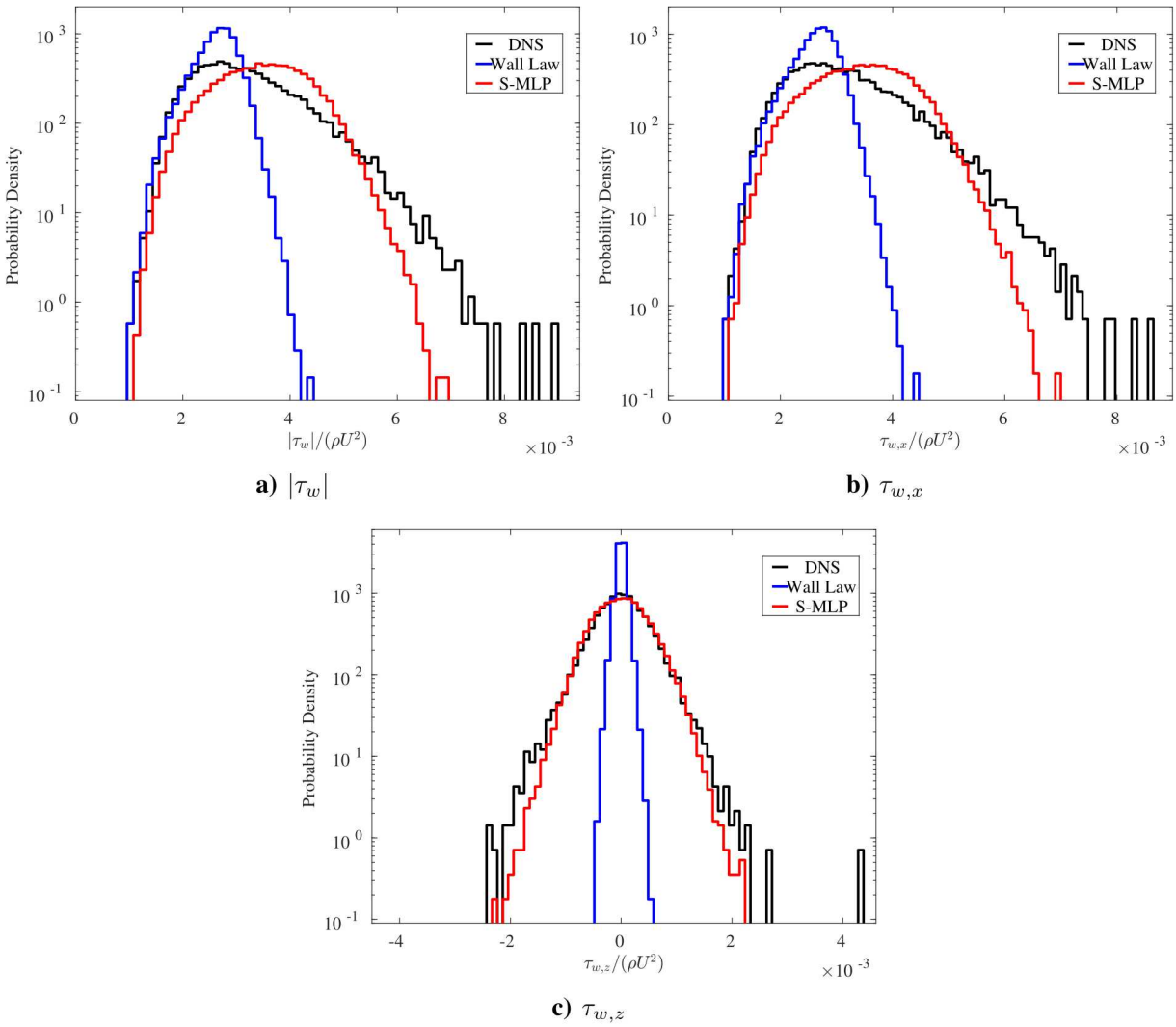

Figure 4: Same as Fig. 3 but for the upstream region of the backward facing step case.

in the domain failed to improve upon the law of the wall predictions (with respect to the MSE) suggests that when the network was trained on a narrower set of data (from only the recirculation region), it was better at recreating the correct $\tau_{w,y}$ values. When a network was trained on the broader set of data which included data from the upstream boundary layer, its ability to predict $\tau_{w,y}$ on the face of the step was lost by the network trying to learn more information.

The PDFs of the wall shear stress magnitudes and its components for each of the prediction methods showed the ability of the invariant representation to predict low-probability stresses of relatively high magnitudes (Fig. 5). Overall, the S-MLP appears to be a significant improvement over the law of the wall, and was perhaps better for this region of this case than for any other that was tested. The spikes in the $\tau_{w,x}$ and $\tau_{w,y}$ PDFs at ≈ 0 come from the vectors taken in locations where individual components are zero by definition. For $\tau_{w,x}$, the zeros come from vectors taken from the step face, while for $\tau_{w,y}$ the zeros come from every location on the floor below the step where the wall-normal shear stress is zero.

Table 3: Same as Table 1 but for the recirculation region for the backward facing step case.

	r	MSE	K-L Div.
$\lvert\tau_{SMLP}\rvert$	0.833	3.49e-7	0.020
$\lvert\tau_{walllaw}\rvert$	0.652	1.14e-6	1.10
$\tau_{SMLP,x}$	0.917	3.22e-7	0.025
$\tau_{walllaw,x}$	0.829	8.35e-7	1.94
$\tau_{SMLP,y}$	0.754	2.96e-9	0.003
$\tau_{walllaw,y}$	0.529	3.66e-9	0.405
$\tau_{SMLP,z}$	0.722	3.11e-7	0.013
$\tau_{walllaw,z}$	0.444	5.40e-7	2.90

Table 4: Same as Table 1 but for the cavity flow case.

	r	MSE	K-L Div.
$\lvert\tau_{SMLP}\rvert$	0.790	1.09e-6	0.138
$\lvert\tau_{walllaw}\rvert$	0.418	2.94e-6	2.47
$\tau_{SMLP,x}$	0.885	7.40e-7	0.160
$\tau_{walllaw,x}$	0.809	1.14e-6	2.36
$\tau_{SMLP,y}$	0.833	3.91e-7	0.023
$\tau_{walllaw,y}$	0.507	1.08e-6	0.605
$\tau_{SMLP,z}$	0.654	6.71e-7	0.012
$\tau_{walllaw,z}$	0.191	1.09e-6	1.08

B. Cavity Flow Case

For the cavity flow case, first a S-MLP ensemble was trained on data from all three regions and then tested on data also taken from all three. Again the S-MLP significantly outperformed the law of the wall in both the quality metrics (Table 4) and in the individual PDFs (Fig. 6).

An issue was again observed with the right-most peak in both the magnitude and in $\tau_{w,x}$ being at a slightly higher value than was seen in the DNS data. This was again attributed to a too-common prediction of values for $\tau_{w,x}$ that were incorrect in the upstream region (as seen in Fig. 4(b)), but also in the downstream region once the attached unidirectional boundary layer was re-established (shown below).

The S-MLP did appear to have an issue with predicting low probability, high positive $\tau_{w,y}$ values. These values primarily come from the upstream "face" of the cavity where recirculating flow is moving in the positive y direction. It may be that the networks were simply not exposed to enough data of this type for the proper behavior to be learned, and it may be that an improved sampling strategy or some other approach may improve performance for these events. Further investigation is warranted.

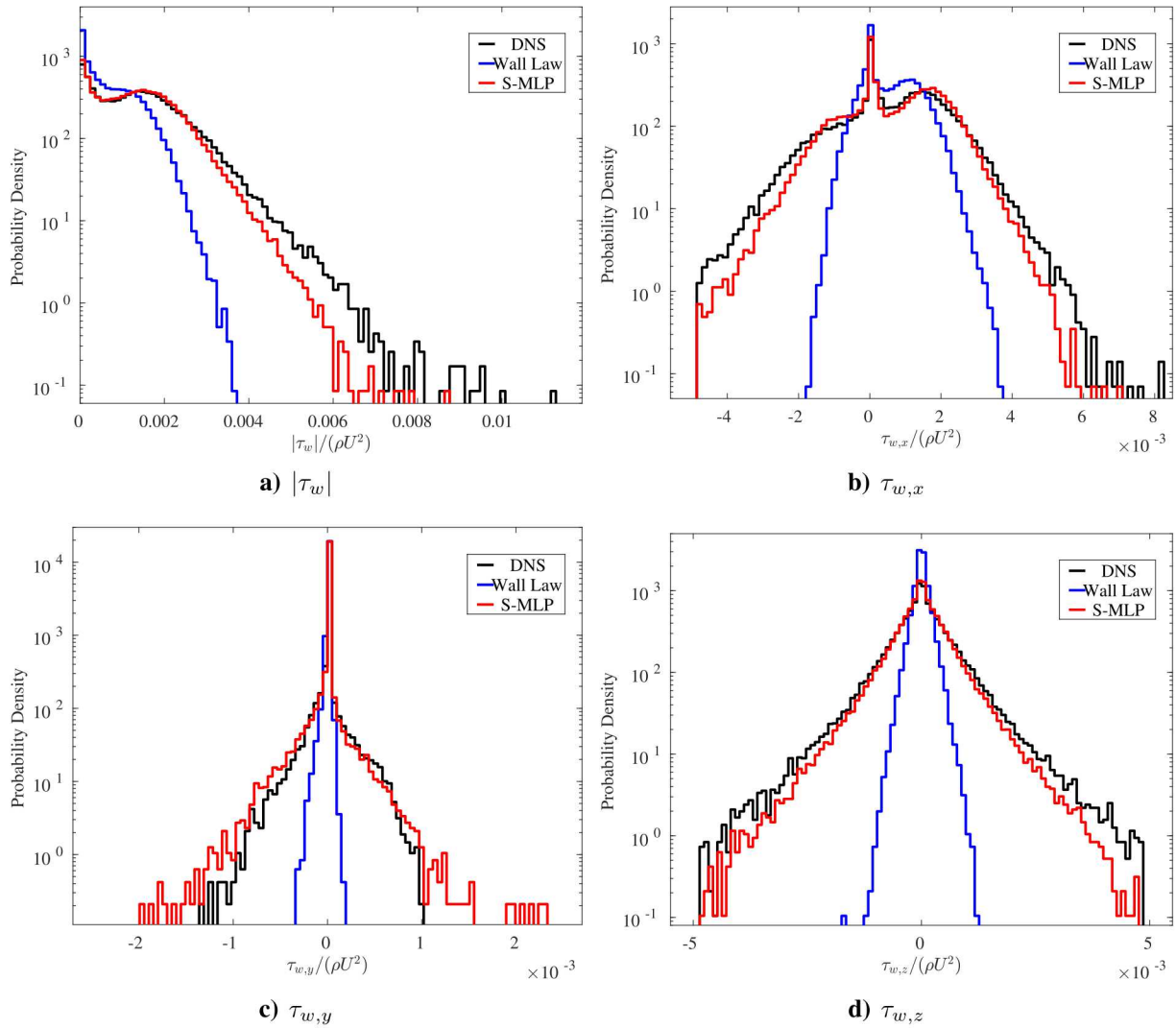

Figure 5: Same as Fig. 3 but for the recirculation region of the backward facing step case.

1. *Upstream Region*

The results for the upstream region of the cavity flow case were nearly indistinguishable from the results for the upstream region of the backward facing step case. Given that the same inlet boundary condition was used in the DNS for both cases and the boundary layer growth rates were observed to be nearly identical upstream of the "step" of the same height used in both cases, it was expected that the wall shear stress information in the two upstream regions would be nearly identical. This being the case, individual PDFs for the magnitude and components of the wall shear stress in the upstream region for the cavity flow case are not shown and the reader is instead referred back to Fig. 4.

2. *Downstream Region*

The S-MLP ensemble network trained and tested on the region downstream of the cavity also showed the same incorrect peak in the PDF of the shear stress magnitude and in $\tau_{w,x}$ (Fig. 7).

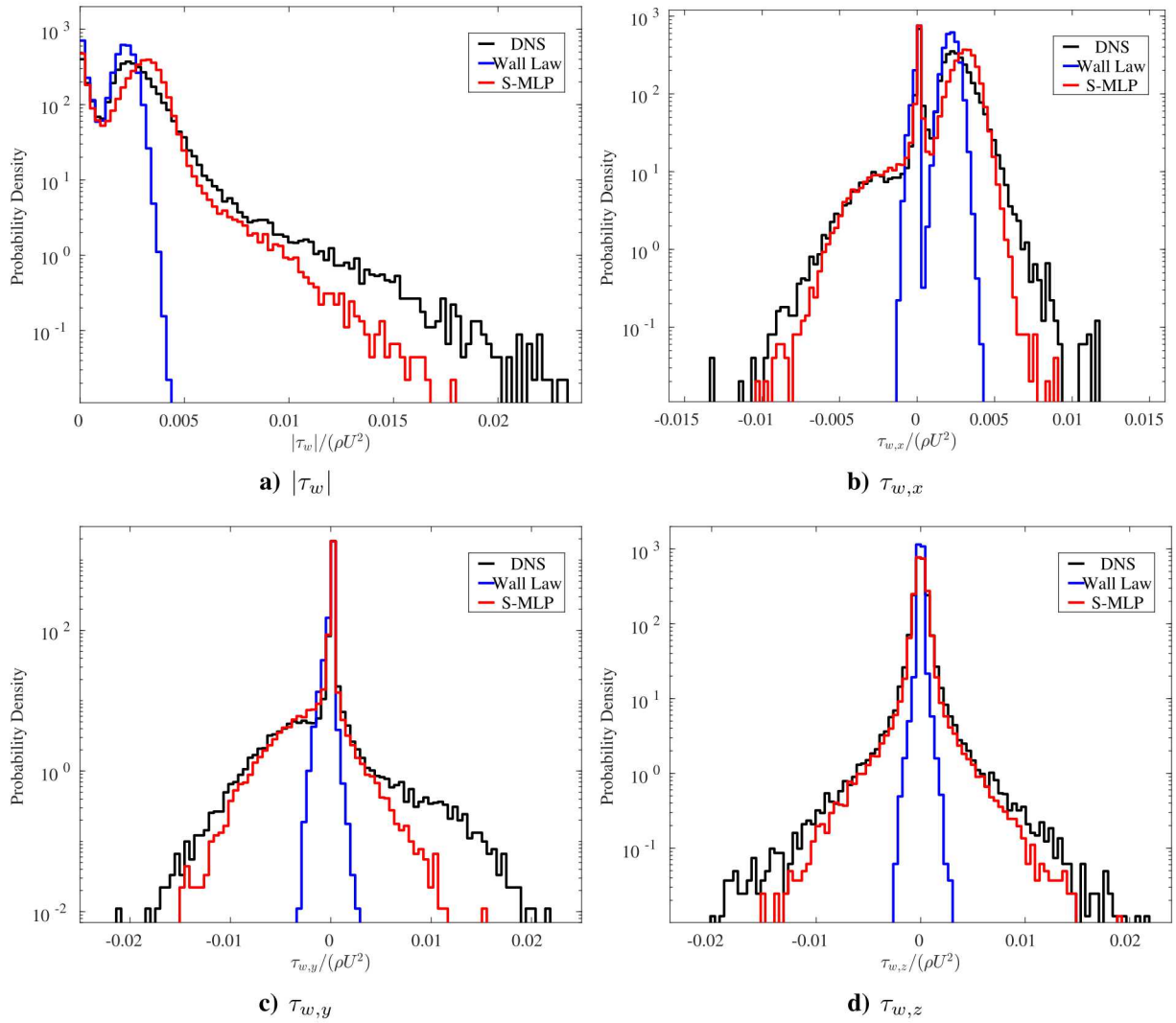

Figure 6: Same as Fig. 3 but for the cavity flow case.

The downstream region contained data from both upstream and downstream of the point at which a traditional unidirectional attached boundary layer was observed along the wall. Upstream of that point (still within the downstream region), the effects of the shear layer impinging at the downstream corner of the cavity resulted in the wall-adjacent flow having a complex topology and even showing recirculation behavior just beyond the cavity lip. This recirculation behavior is what causes negative $\tau_{w,x}$ values in the downstream region, while the attached boundary layer seen downstream of its re-establishment point was responsible for the positive $\tau_{w,x}$ values. Because of the similarity between that downstream attached boundary layer and the boundary layer behavior in the upstream region, the same incorrect peak is seen in the PDF of $\tau_{w,x}$ predictions in the downstream region (Fig. 7(b)) as was seen in the upstream region (Fig. 4(b)).

Regardless of the slightly missed peak for $\tau_{w,x}$, the quality metrics showed that the S-MLP still did better than the law of the wall for both the magnitude and $\tau_{w,x}$ (Table 5). The behavior for $\tau_{w,z}$ was perhaps the most interesting, in that the law of the wall predictions had a near zero r, but also had a slightly better MSE than the S-MLP. The low correlation produced by law of the wall for $\tau_{w,z}$ can be attributed to the assumption of alignment between the wall shear stress and the velocity

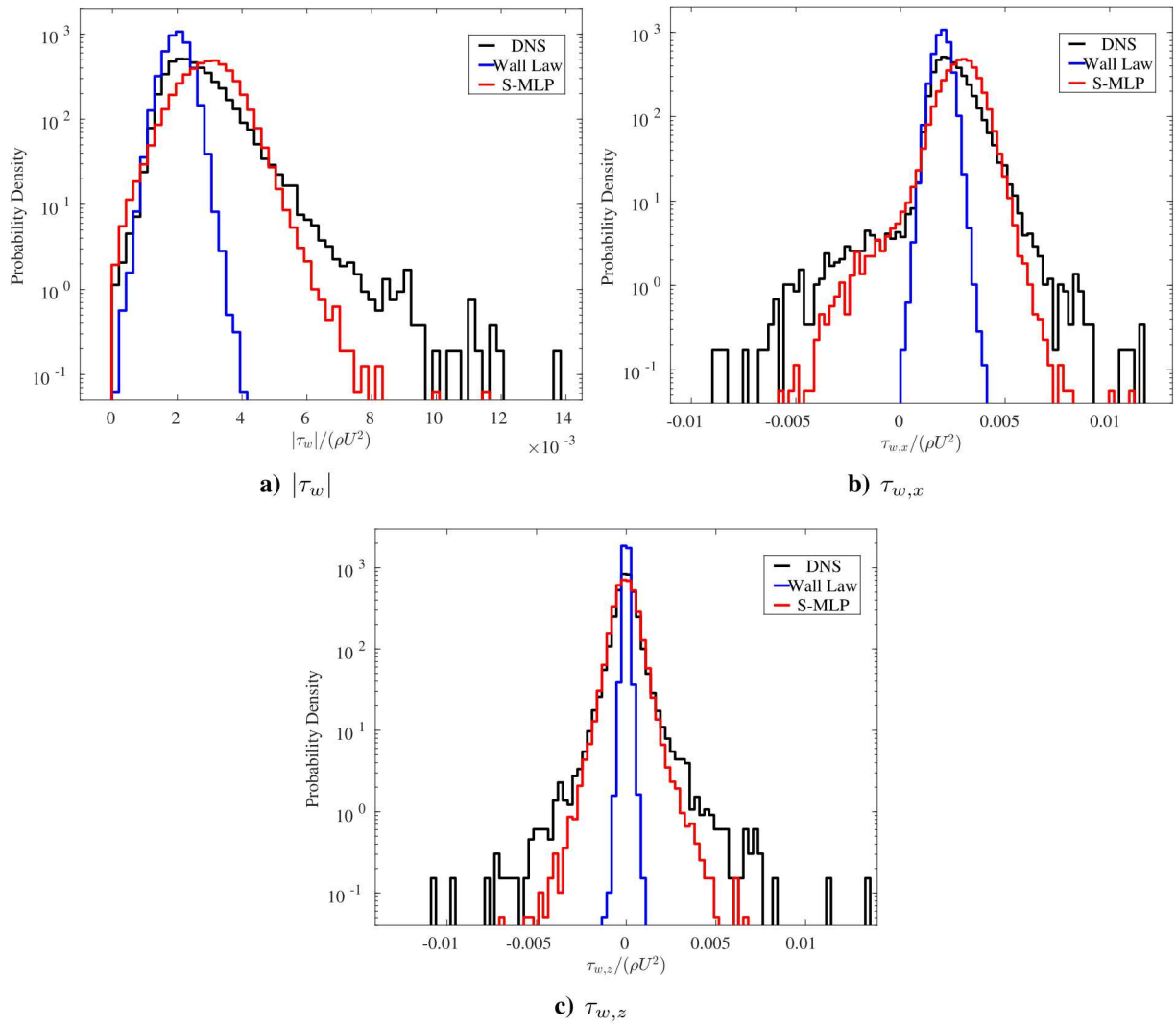

Figure 7: Same as Fig. 3 but for the downstream region of the cavity case.

vector used in the law-of-the-wall model. This assumption is not consistent with the filtered DNS data, resulting in a very low correlation for the spanwise stress component (see [22]). The law of the wall also consistently predicted lower magnitudes for $\tau_{w,z}$ than were correct, leading to an MSE that was only slightly better than if $\tau_{w,z}$ was assumed to be zero everywhere in the region. The S-MLP more often predicted larger magnitude values and produced a much improved correlation with the filtered DNS. However, many of the individual predicted values from the S-MLP must have had larger errors than if something close to zero had been predicted, thus increasing the MSE on the whole.

3. *Cavity Region*

For the cavity region of the cavity flow case, again the S-MLP ensemble network trained on data from solely that region significantly outperformed the law of the wall in predicting the shear stresses. This improvement was evident in all of the quality metrics for all of the components (Table 6), as well as in the PDFs and the corresponding K-L Divergences (Fig. 8). The distribution

Table 5: Same as Table 1 but for the downstream region of the cavity case.

	r	MSE	K-L Div.
$\lvert\tau_{SMLP}\rvert$	0.476	9.82e-7	0.217
$\lvert\tau_{walllaw}\rvert$	0.337	1.24e-6	2.52
$\tau_{SMLP,x}$	0.533	1.10e-6	0.220
$\tau_{walllaw,x}$	0.280	1.32e-6	2.47
$\tau_{SMLP,z}$	0.410	5.27e-7	0.050
$\tau_{walllaw,z}$	0.097	5.11e-7	2.99

Table 6: Same as Table 1 but for the cavity region of the cavity flow case.

	r	MSE	K-L Div.
$\lvert\tau_{SMLP}\rvert$	0.862	1.71e-6	0.051
$\lvert\tau_{walllaw}\rvert$	0.306	7.57e-6	4.17
$\tau_{SMLP,x}$	0.873	3.41e-7	0.050
$\tau_{walllaw,x}$	0.694	1.18e-6	5.46
$\tau_{SMLP,y}$	0.834	1.35e-6	0.088
$\tau_{walllaw,y}$	0.501	3.79e-6	2.25
$\tau_{SMLP,z}$	0.724	1.45e-6	0.028
$\tau_{walllaw,z}$	0.221	2.85e-6	1.70

of $\tau_{w,x}$ was concentrated towards negative values, all coming from the circulating flow along the floor of the cavity. Unlike the recirculation region of the backward facing step flow, there was not a peak at positive $\tau_{w,x}$ values because no reattachment occurs.

Of all of the individual regions tested throughout this work, this region had the highest shear stress magnitudes, driven primarily by the most extreme individual values of $\tau_{w,y}$ and $\tau_{w,z}$ observed in any region studied here. These extreme values were primarily found on the downstream "face" of the cavity where the shear layer generated by the flow separation at the upstream edge impinged on the downstream cavity corner. Unlike the law of the wall, the S-MLP had no problem capturing and recreating these more extreme values, suggesting that the S-MLP may be a promising approach to modeling cavity flow and other complex flows with regions of separation.

C. Predicting the Backward Facing Step with Cavity Flow Case Data

The final test reported here is one in which the S-MLP ensemble that was trained on data from all three regions of the cavity flow case was tested on the 20% of the data taken collectively from the two regions of the backward facing step case. Because the cavity flow case included all of the same flow phenomena (boundary layer, recirculation, reattachment, etc) observed in the backward facing step case, and the cavity flow case had more extreme values on which the S-MLPs were trained, it was hypothesized that this network ensemble would be able to accurately predict the backward facing step data. The PDFs (Fig. 9) and quality metrics (Table 7) all confirmed this expectation. The S-MLP ensemble trained on the cavity flow data was a much better predictor of the wall shear stresses from the backward facing step case than was the law of the wall, and was only slightly

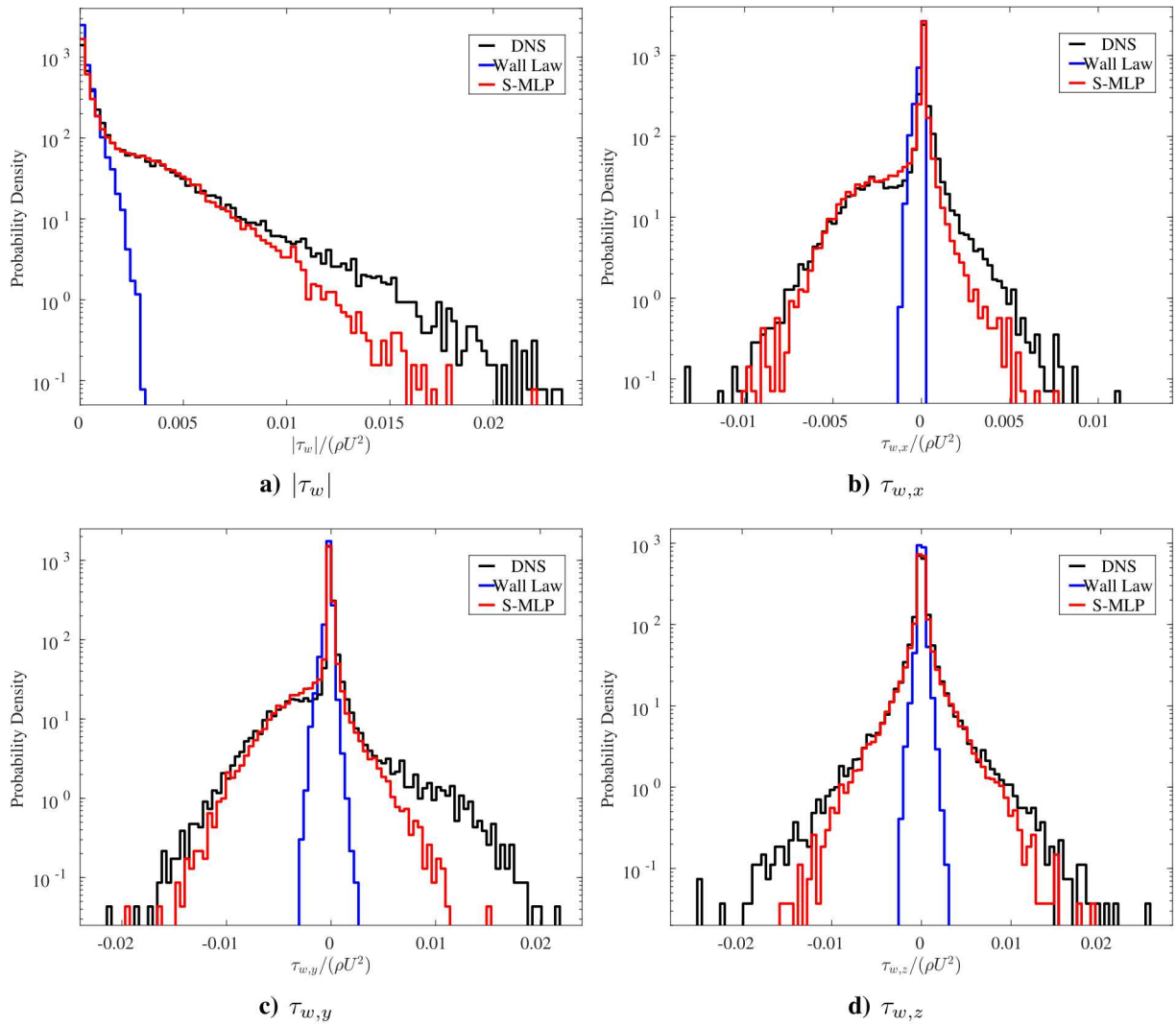

Figure 8: Same as Fig. 3 but for the cavity region of the cavity flow case.

worse than was the network ensemble trained on the backward facing step data. In fact, for $\tau_{w,y}$ specifically, this network outperformed the network trained on backward facing step data, both in MSE and KLD (compare to Table 1). This suggests that the cavity-trained network was better at predicting the extreme low-probability events of $\tau_{w,y}$ than was the backward-facing-step-trained network.

VI. Discussion, Conclusions, and Future Work

The results of the neural network near-wall models for wall shear stress gave reasonable predictions in the upstream regions of the two flow cases, but it is surely the performance of the models in the recirculation and cavity regions which was the most striking. Modern near-wall models have often demonstrated acceptable accuracy for attached boundary layer flows, and thus improvements upon shear stress predictions in those regions may be of minimal impact. But it is situations such as the backward facing step recirculation and cavity regions that prove particularly challenging for near-wall turbulence models.

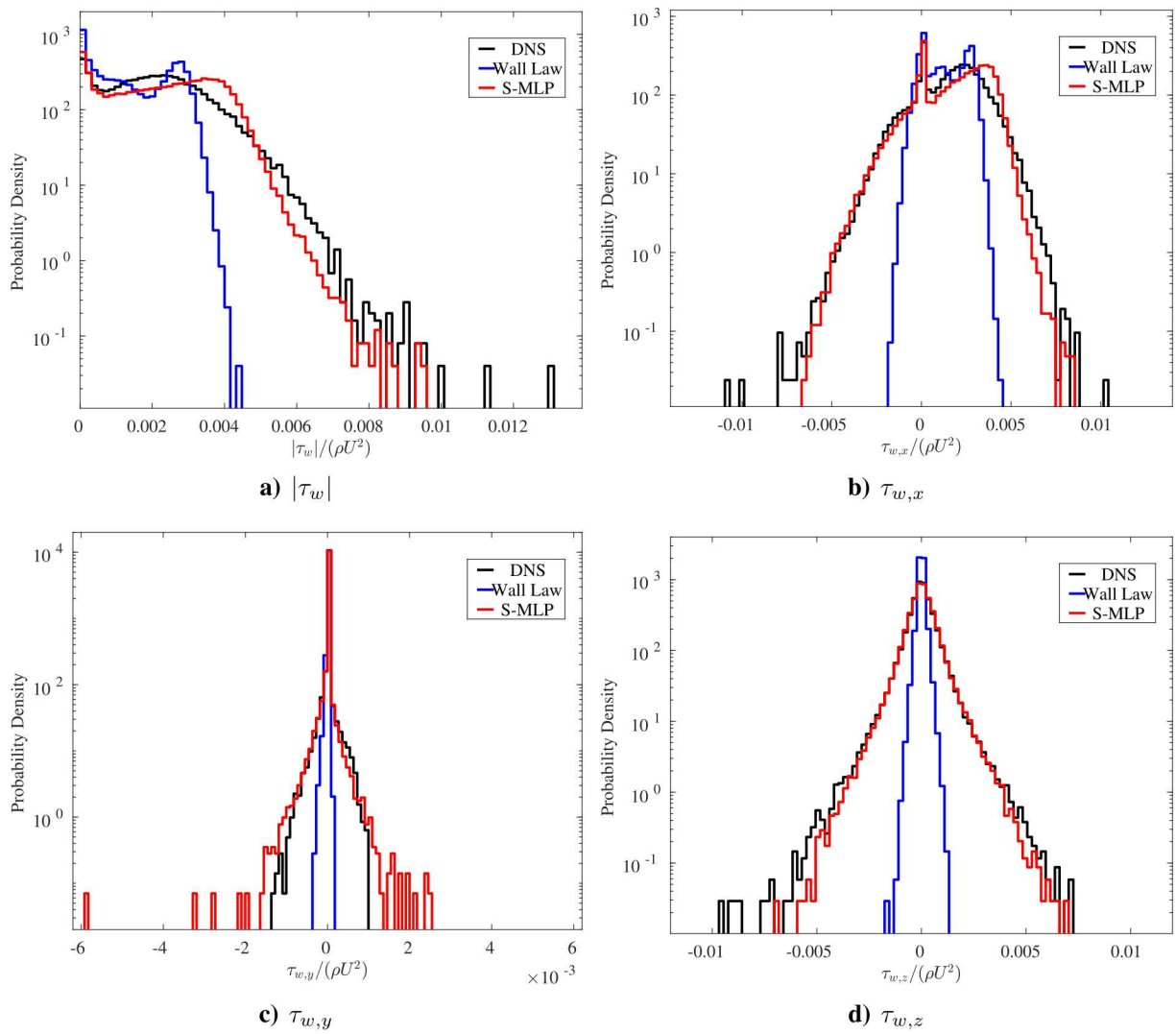

Figure 9: Same as Fig. 3 but for the S-MLPs trained on data from the cavity flow case and tested on data from the backward facing step case.

Based on the *a priori* results reported here, the S-MLP implementation has the potential to significantly improve wall shear stress modeling in separated flows. This includes improvements to both the accuracy of individual predictions, as well as for the statistical properties of wall stress distributions. It should be noted that it is currently unknown how well this approach will work when implemented and tested in a LES code. One known issue is that wall-modeled LES dynamics will differ from those found in filtered DNS, due to the use of imperfect subgrid and near-wall models in the LES. The ability to use a trained S-MLP as the wall shear stress function has been implemented into an in-house LES code at Sandia National Laboratories. Preliminary testing on a channel flow case has demonstrated that a trained network can be used as a wall shear stress function within the code. Further testing of the accuracy of the predicted stresses, as well as the accuracy of the associated flow-field statistics within the channel, as well as confirmation of the expected coordinate invariance property of the models, is ongoing.

One other limitation of the work reported here was that we only considered specific sets of

Table 7: Same as Table 1 but for the S-MLPs trained on data from the cavity flow case and tested on data from the backward facing step case.

	r	MSE	K-L Div.
$\lvert\tau_{SMLP}\rvert$	0.842	6.58e-7	0.072
$\lvert\tau_{walllaw}\rvert$	0.782	1.11e-6	1.75
$\tau_{SMLP,x}$	0.904	7.02e-7	0.071
$\tau_{walllaw,x}$	0.875	9.05e-7	2.11
$\tau_{SMLP,y}$	0.689	2.09e-9	0.001
$\tau_{walllaw,y}$	0.528	2.31e-9	0.258
$\tau_{SMLP,z}$	0.655	3.22e-7	0.004
$\tau_{walllaw,z}$	0.394	4.23e-7	2.29

filter sizes, flow variables, and network hyperparameters and it may be that some other set or combination of sets would be capable of producing even better (or obviously worse) results. Some testing of the number of layers and the number of nodes within each layer, along with the number of networks to use in each ensemble has been done. Further testing of these values is ongoing in order to determine the optimal choices that produce the best results while minimizing network complexity and training times. The values chosen here were picked somewhat arbitrarily, and although better results were found with these values than with smaller networks/ensembles (*e.g.*, three hidden layers with fewer nodes and only three networks in the ensemble), it is not yet known whether results could be further improved with different parameter values or if the network could be reduced in complexity while still returning results of equivalent quality.

The flow variables chosen here, the choice of which dictated the size and shape of the representation, were picked because of their availability, universality, applicability, and simplicity. It may be the case however that some of the data used here is unnecessary (*e.g.*, because of redundancy) and/or that the inclusion of some other flow features may improve the results farther. Near-wall pressures and pressure gradients, for example, are expected to be important for near-wall modeling in some flow cases and their inclusion is yet to be tested. If the pressure at the wall and the near-wall pressure gradient vector were included in the input feature set, it would result in a larger set of λ's and Π's and thus larger networks, but may produce superior results.

The results of a network trained on one set of data and tested on data from a somewhat different flow case are also promising for the potential portability and applicability of this approach. Testing of this possibility is also ongoing. Ideally the objective would be to produce a single, optimally-designed network, which was trained on a suite of flow cases that could be called upon as the near-wall function in operational LES codes and which would be an accurate predictor of wall conditions even for flow cases on which it was never trained.

References

[1] Zhang, Z. J. and Duraisamy, K., "Machine learning methods for data-driven turbulence modeling," *AIAA Paper 2015-2460*, June 2015.

[2] Ling, J., Kurzawski, A., and Templeton, J., "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance," *J. Fluid Mech.*, Vol. 807, 2016, pp. 155–166.

[3] Wang, J.-X., Wu, J.-L., and Xiao, H., "Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data," *Phys. Rev. Fluids*, Vol. 2, 2017, pp. 034603.

[4] Ling, J., Barone, M., Davis, W., Chowdhary, K., and Fike, J., "Development of Machine Learning Models for Turbulent Wall Pressure Fluctuations," January 2017.

[5] Milano, M. and Koumoutsakos, P., "Neural Network Modeling for Near Wall Turbulent Flow," *J. Comp. Phys.*, Vol. 182, 2002, pp. 1–26.

[6] Larsson, J., Kawai, S., Bodart, J., and Bermejo-Moreno, I., "Large eddy simulation with modeled wall-stress: Recent progress and future directions," *Mechanical Engineering Reviews, JSME*, Vol. 3, 2016, pp. 15–00418.

[7] Piomelli, U. and Balaras, E., "Wall-Layer Models for Large-Eddy Simulations," *Ann. Rev. Fluid Mech.*, Vol. 34, 2002, pp. 349–374.

[8] Spencer, A. J. M., "Part III: Theory of Invariants," *Continuum Physics Volume I*, edited by A. C. Eringen, Academic Press, 1971, pp. 239–353.

[9] Zheng, Q.-S. and Boehler, J. P., "The description, classification, and reality of material and physical symmetries," *Acta Mechanica*, Vol. 102, 1994, pp. 73–89.

[10] Zheng, Q.-S., "On transversely isotropic, orthotropic and relative isotropic functions of symmetric tensors, skew-symmetric tensors and vectors. Part I: Two dimensional orthotropic and relative isotropic functions and three dimensional relative isotropic functions," *Int. J. Engng Sci.*, Vol. 31, 1993, pp. 1399–1409.

[11] Zheng, Q.-S., "Theory of representations for tensor function–A unified invariant approach to constitutive equations," *Appl Mech Rev.*, Vol. 47, 1994, pp. 545–587.

[12] Zheng, Q.-S., "On transversely isotropic, orthotropic and relative isotropic functions of symmetric tensors, skew-symmetric tensors and vectors. Part II: The representations for three dimensional transversely isotropic functions," *Int. J. Engng Sci.*, Vol. 31, 1993, pp. 1411–1423.

[13] Spencer, A. J. M., "Isotropic polynomial invariants and tensor functions," *Applications of tensor functions in solid mechanics*, edited by J. P. Boehler, Springer-Verlag Wien, 1987, pp. 141–169.

[14] Arunajatesan, S., Ross, M., and Garrett, T. J., "Validation of an FSI Modeling Framework for Internal Captive Carriage Applications," *AIAA Paper 2013-2157*, May 2013.

[15] Barone, M. and Arunajatesan, S., "Pressure loading within rectangular cavities with and without a captive store," *J. of Aircraft*, Vol. 53, No. 4, 2016, pp. 982–991.

[16] Diaz-Daniel, C., Laizet, S., and Vassilicos, J. C., "Wall shear stress fluctuations: Mixed scaling and their effects on velocity fluctuations in a turbulent boundary layer," *Phys. of Fluids*, Vol. 29, 2017, pp. 055102.

[17] Zheng, L., Duffner, S., Idrissi, K., Garcia, C., and Baskurt, A., "Siamese multi-layer perceptrons for dimensionality reduction and face identification," *Multimed Tools Appl*, Vol. 75, 2016, pp. 5055–5073.

[18] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., andI. Goodfellow, S. G., Harp, A., Irving, G., M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F.,

Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015, Software available from tensorflow.org.

[19] Seide, F. and Agarwal, A., "CNTK: Microsoft's Open-Source Deep-Learning Toolkit," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, ACM, New York, NY, USA, 2016, pp. 2135–2135.

[20] Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., Bastien, F., Bayer, J., Belikov, A., Belopolsky, A., Bengio, Y., Bergeron, A., Bergstra, J., Bisson, V., Bleecher Snyder, J., Bouchard, N., Boulanger-Lewandowski, N., Bouthillier, X., de Brébisson, A., Breuleux, O., Carrier, P.-L., Cho, K., Chorowski, J., Christiano, P., Cooijmans, T., Côté, M.-A., Côté, M., andY. N. Dauphin, A. C., Delalleau, O., Demouth, J., Desjardins, G., Dieleman, S., Dinh, L., Ducoffe, M., Dumoulin, V., Ebrahimi Kahou, S., Erhan, D., Fan, Z., Firat, O., Germain, M., Glorot, X., Goodfellow, I., Graham, M., Gulcehre, C., Hamel, P., Harlouchet, I., Heng, J.-P., Hidasi, B., Honari, S., Jain, A., Jean, S., Jia, K., Korobov, M., Kulkarni, V., Lamb, A., Lamblin, P., Larsen, E., Laurent, C., Lee, S., Lefrancois, S., Lemieux, S., Léonard, N., Lin, Z., Livezey, J. A., Lorenz, C., Lowin, J., Ma, Q., Manzagol, P.-A., Mastropietro, O., McGibbon, R. T., Memisevic, R., van Merriënboer, B., Michalski, V., Mirza, M., Orlandi, A., Pal, C., Pascanu, R., Pezeshki, M., Raffel, C., Renshaw, D., Rocklin, M., Romero, A., Roth, M., Sadowski, P., Salvatier, J., Savard, F., Schlüter, J., Schulman, J., Schwartz, G., Serban, I. V., Serdyuk, D., Shabanian, S., Simon, E., Spieckermann, S., Subramanyam, S. R., Sygnowski, J., Tanguay, J., van Tulder, G., Turian, J., Urban, S., Vincent, P., Visin, F., de Vries, H., Warde-Farley, D., Webb, D. J., Willson, M., Xu, K., Xue, L., Yao, L., Zhang, S., and Zhang, Y., "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, Vol. abs/1605.02688, May 2016.

[21] Spalding, D. B., "A Single Formula for the "Law of the Wall?" *J. Appl. Mechanics*, Vol. 28, 1961, pp. 455–458.

[22] Chin, C. C., Monty, J. P., Ooi, A. S. H., and Marusic, I., "Investigation of velocity correlations in turbulent channel flow," *16th Australasian Fluid Mechanics Conference*, Gold Coast, Queensland, Australia, 2007.