SAND2019-1703C

Coordinate-Invariant Near-Wall Turbulence Modeling via Neural Networks

PRESENTED BY

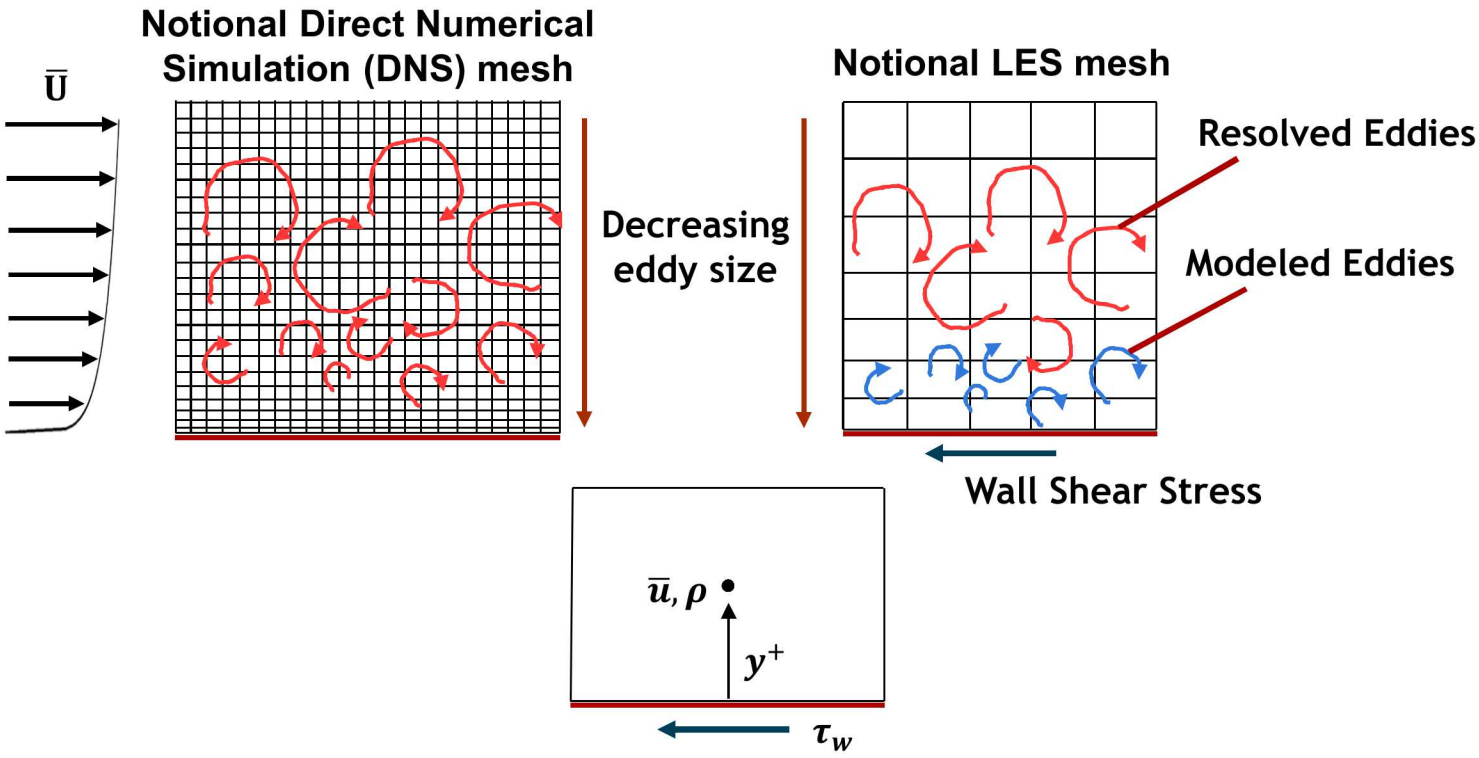
Nathan E Miller
Matthew Barone, Warren L Davis, Jeffrey A Fike

Sandia National Laboratories
Albuquerque, NM

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & 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.

Data-Driven Near-Wall Modeling

Near-wall turbulence modeling for Large Eddy Simulation (LES):



Typically a solution to the mean flow equations at each near-wall grid cell

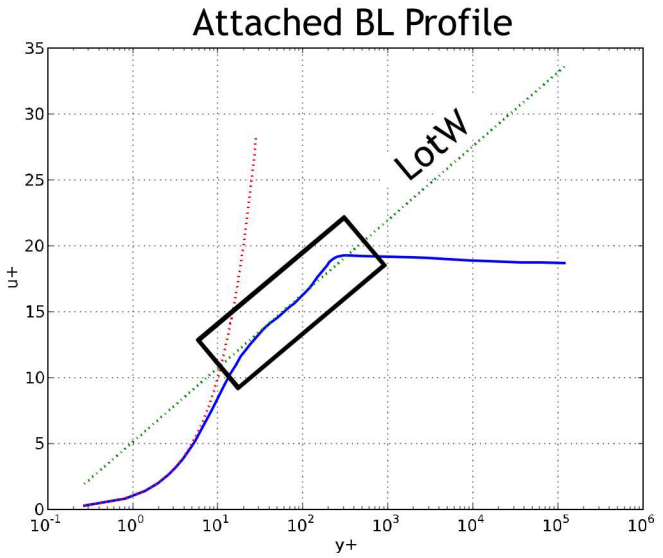
Law of the Wall (LotW):

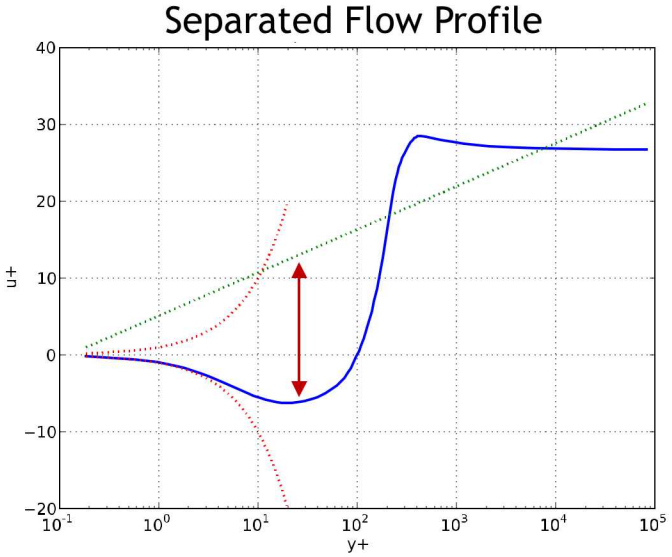
$$\frac{\bar{u}}{u_\tau} = \frac{1}{k}\ln(y^+) + b \quad \text{where} \quad u_\tau = \sqrt{\frac{\tau_w}{\rho}}$$

Data-Driven Near-Wall Modeling

$$\frac{\bar{u}}{u_\tau} = \frac{1}{k}\ln(y^+) + b \quad \text{where} \quad u_\tau = \sqrt{\frac{\tau_w}{\rho}}$$

- Assumes grid cell y^+ is within log layer
- Assumes mean boundary layer profile is appropriate
  - Fine for large cells + attached flows
  - Not true for separated flows

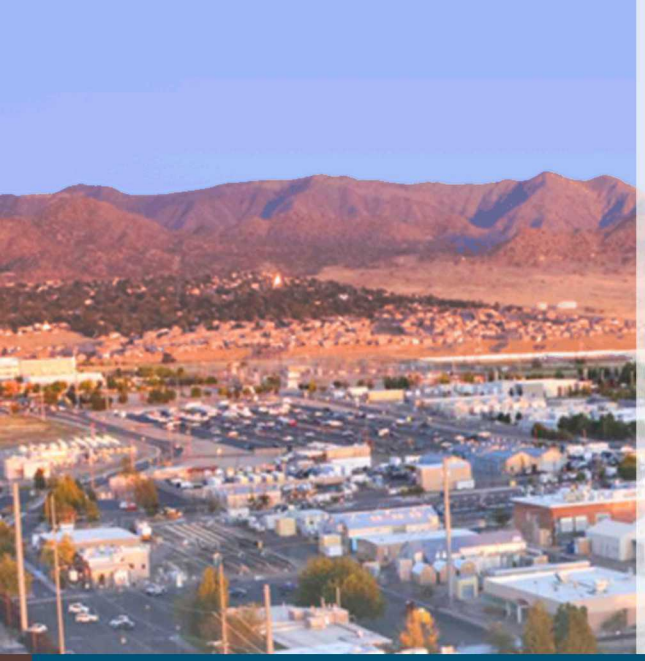




The Goal: A general expression for wall shear stress based on filtered state variables at the near-wall LES grid cell

$$\tau_{w,i} = F(\vec{\mathbf{n}}, \bar{\mathbf{U}}, \bar{\mathbf{S}}, \bar{\mathbf{\Omega}})$$

- At least as good as LotW for attached flows
- Better than LotW for detached flows



Invariant Theory Approach

Invariant Theory Approach

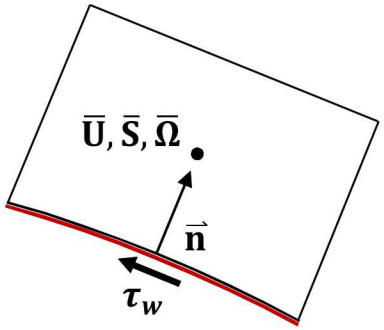
We seek a general coordinate frame invariant representation for the wall shear stress:

$$\tau_{w,i} = F(\vec{\mathbf{n}}, \bar{\mathbf{U}}, \bar{\mathbf{S}}, \bar{\mathbf{\Omega}})$$

Invariant theory presents as a tensor polynomial of **Form-Invariants**

$$\tau_{w,i} = \sum_k G^{(k)} \Pi_i^{(k)}$$

$$\Pi^{(k)} = f(\vec{\mathbf{n}}, \bar{\mathbf{U}}, \bar{\mathbf{S}}, \bar{\mathbf{\Omega}})$$



Because $\boldsymbol{\tau}$ is a vector, $\boldsymbol{\Pi}$'s must be vectors.

Coefficients based on unknown functions of **Scalar Invariants**

$$G^{(k)} = \mathcal{F}^{(k)}(\lambda_1, \lambda_2, \lambda_3, \ldots \lambda_n)$$

$$\lambda_j = f(\vec{\mathbf{n}}, \bar{\mathbf{U}}, \bar{\mathbf{S}}, \bar{\mathbf{\Omega}})$$

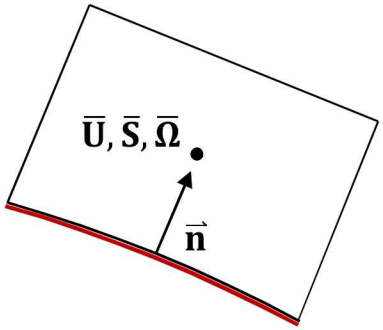
Put Together:

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

The Invariants

Given one symmetric tensor ($\bar{\mathbf{S}}$), one skew-symmetric tensor ($\bar{\mathbf{\Omega}}$), & two vectors ($\bar{\mathbf{U}}, \vec{\mathbf{n}}$)

- Invariant to rotation about $\vec{\mathbf{n}}$
- 20 Scalar Invariants: $\lambda_j = f(\vec{\mathbf{n}}, \bar{\mathbf{U}}, \bar{\mathbf{S}}, \bar{\mathbf{\Omega}})$ *(Zheng 1993a, Zheng 1994)*



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

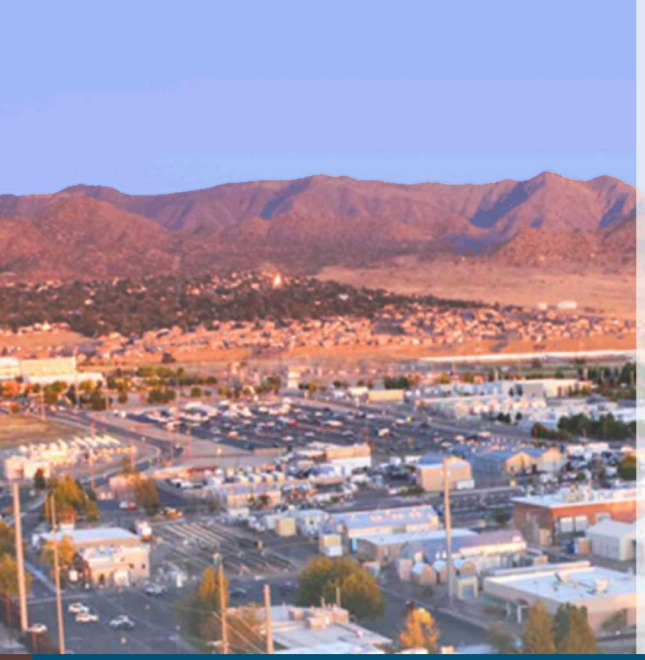
- Form-Invariants derived from Scalar Invariants *(Spencer 1987)*
- 8 Form-Invariants: $\Pi^{(k)} = f(\vec{\mathbf{n}}, \bar{\mathbf{V}}, \bar{\mathbf{S}}, \bar{\mathbf{\Omega}})$ *(Zheng 1993b, Zheng 1994)*

$$\begin{array}{ll}
\Pi^{(1)} = \bar{\mathbf{U}}, & \Pi^{(2)} = \epsilon\bar{\Omega}, \\
\Pi^{(3)} = \mathbf{n}, & \Pi^{(4)} = \bar{\mathbf{S}}\mathbf{n}, \\
\Pi^{(5)} = \mathbf{n} \times \bar{\mathbf{S}}\mathbf{n}, & \Pi^{(6)} = \bar{\mathbf{\Omega}}\mathbf{n}, \\
\Pi^{(7)} = \mathbf{n} \times \bar{\mathbf{U}}, & \Pi^{(8)} = \mathbf{n} \times \bar{\mathbf{\Omega}}\mathbf{n}
\end{array}$$

$$\tau_{w,i} = \sum_{k=1}^{8} G^{(k)} \Pi_i^{(k)}$$

$$G^{(k)} = \mathcal{F}^{(k)}(\lambda_1, \lambda_2, \lambda_3, \ldots \lambda_{20})$$

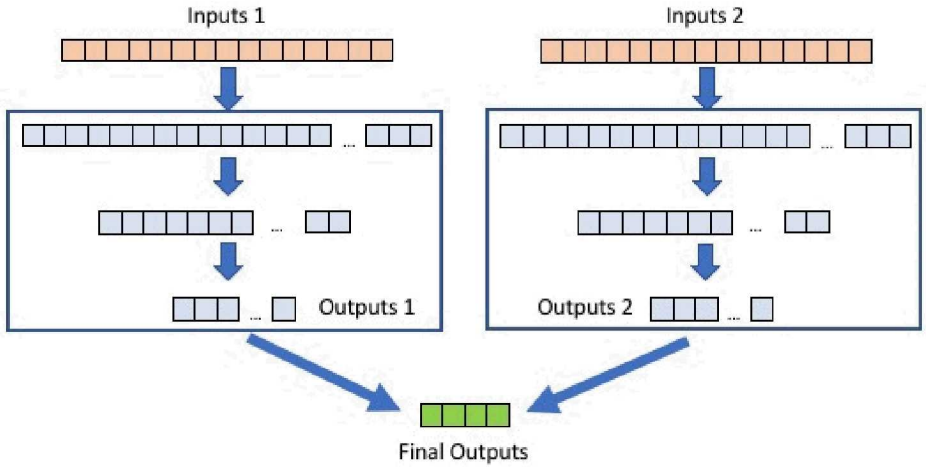
- ML model that can learn $\mathcal{F}^{(k)}$ to get $G^{(k)}$ to combine with $\Pi_i^{(k)}$ to predict $\tau_{w,i}$



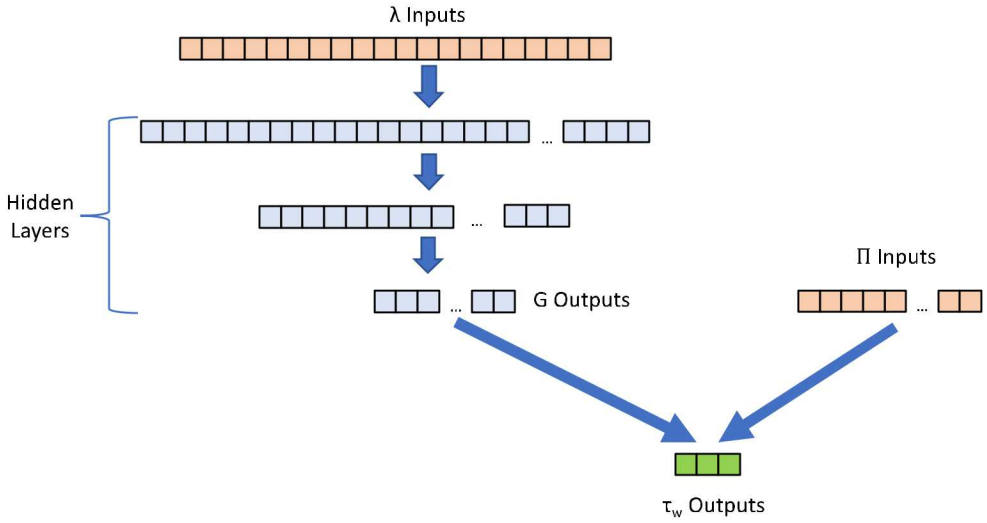
Neural Network Approach

Siamese Multi-Layer Perceptron

Two parallel MLPs of equivalent architecture with the same weights (Zheng 2016)



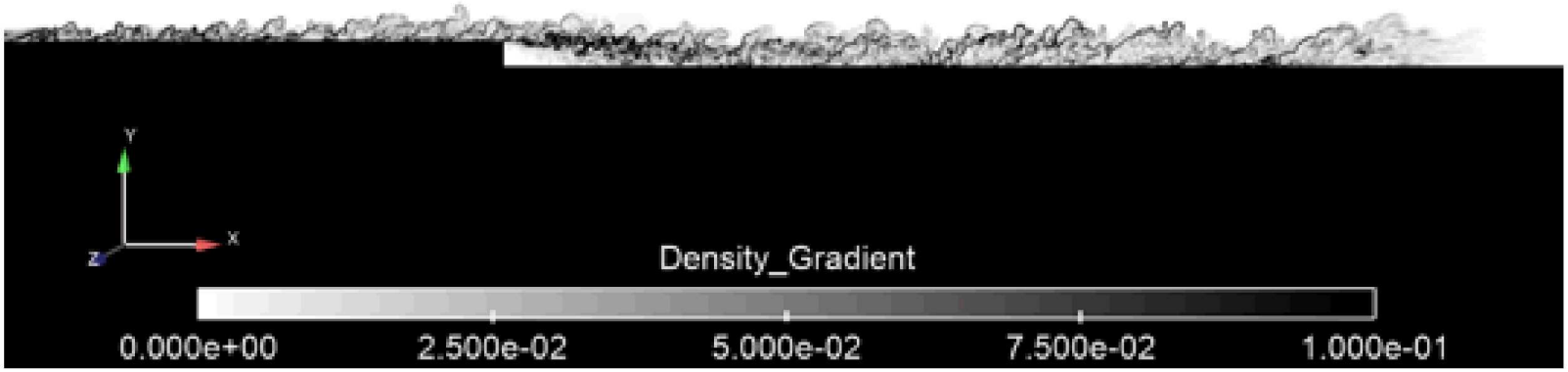
- The functions $G^{(k)} = \mathcal{F}^{(k)}(\lambda_j)$ must be learned
- Only need to train one side of our SMLP architecture
- Combination of open-source:
  Keras, TensorFlow
  and Sandia proprietary ML tools:
  TensorNet, Groupy
- 4 hidden layers, ≈5100 weights
- 3x multiplier, ReLU, Mean error loss func



DNS Data and LES filtering

Backwards Facing Step DNS

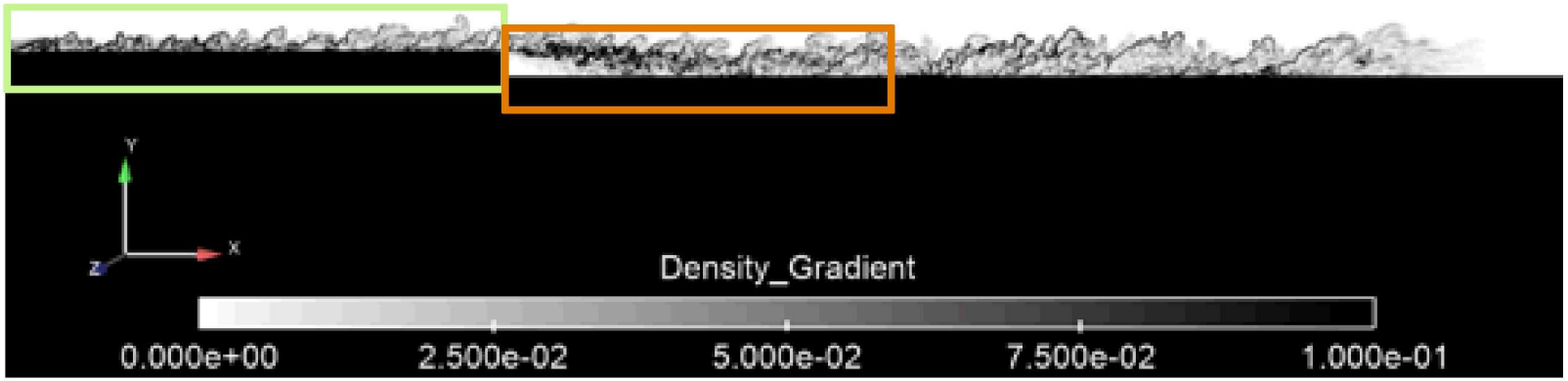
- SNL's structured, multi-block, finite volume code: SIGMA-CFD *(Arunajatesan et al. 2013)*
- Mach 0.6 compressible freestream
- Reynolds Number = 6254
- Inlet condition from precursor RANS simulation
- Boundary layer height increased to $3\delta_0$ at the step
- Step height also $3\delta_0$ $\Rightarrow$ $h/\delta = 1.0$
- $97\delta_0 \times 120\delta_0 \times 6\delta_0$ upstream
- $130\delta_0 \times 120\delta_0 \times 6\delta_0$ downstream
- $\approx$ 237 million grid cells with stretching in vertical and streamwise
  - Points concentrated near wall, near step, in shear layer, and near reattachment



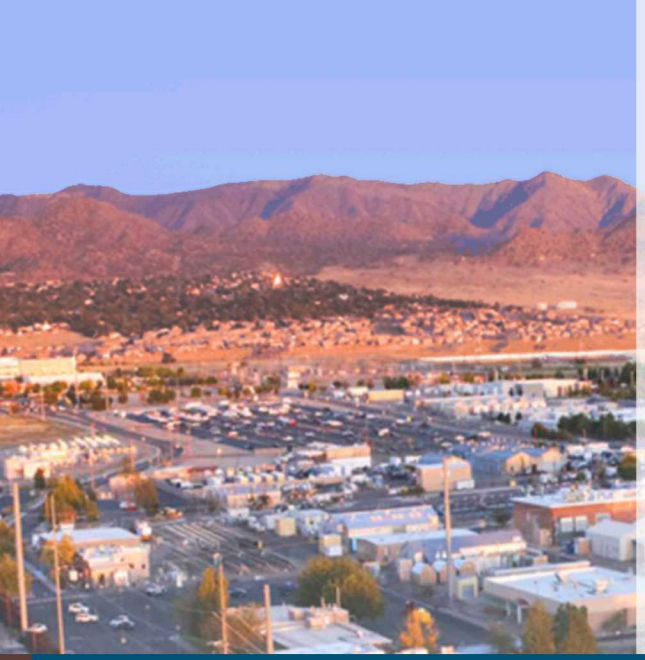
Pseudo-LES Filtering and Selective Sampling

- Upstream Boundary Layer Region:
  - 5800 pseudo-LES grid cells
  - $y^+ = [30,60,100,150]$
  - $[2\Delta_x^+, 2\Delta_z^+] = [80,50]$ & $[170,110]$ (Larsson *et al.* 2016)

- Downstream Detached Region:
  - 900 pseudo-LES grid cells on step face
  - 3660 pseudo-LES grid cells on "floor"
  - $y^+ = [15,30,60]$ from downstream attached BL
  - 8×8 DNS grid cells



64-16-20 split. Selective sampling on training data. Ensemble of 10 SMLPs.



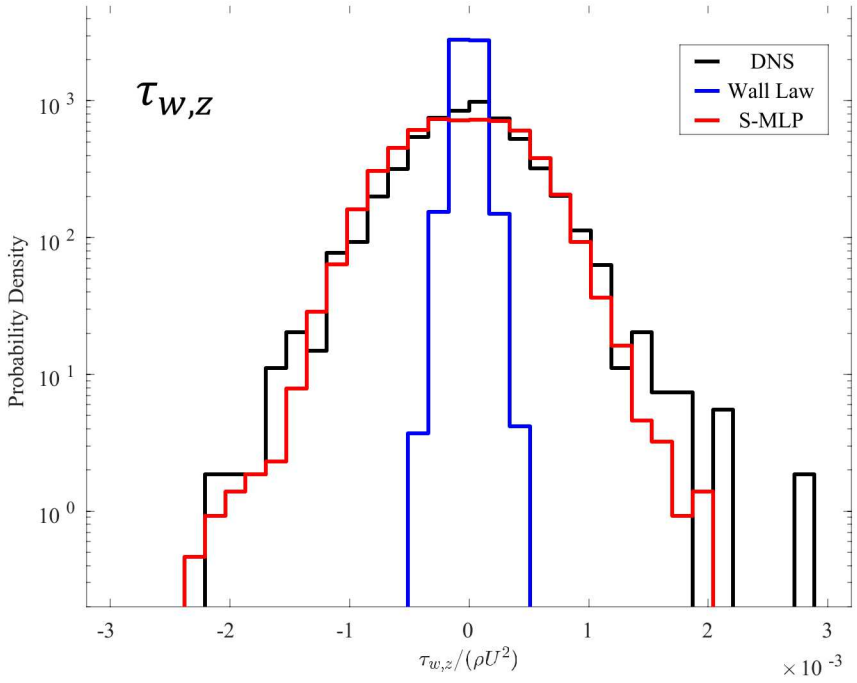
Results

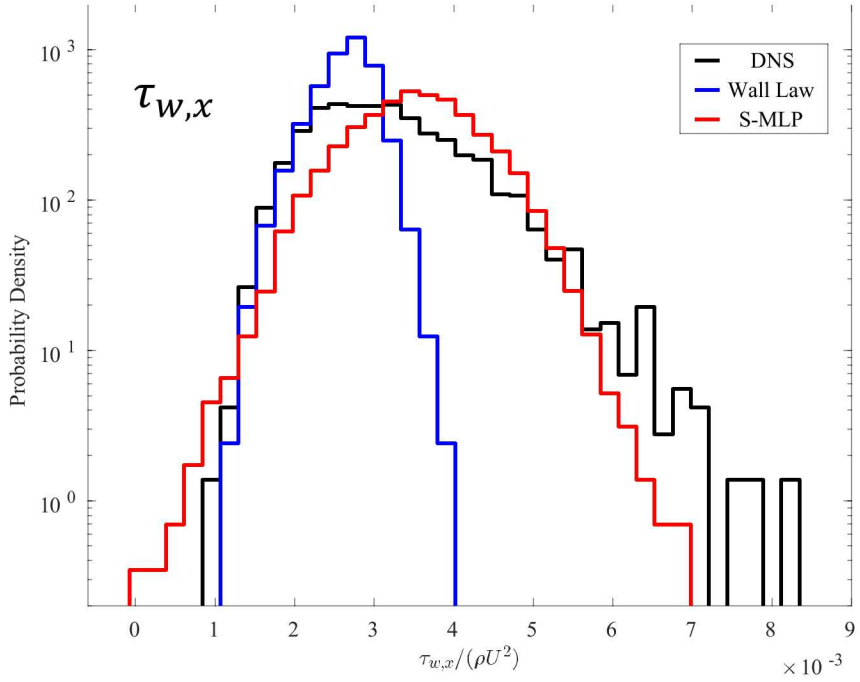
Quality Metrics

- Compare SMLP and Law of the Wall (LotW) predictions to true DNS stress values
- Quality metrics, predicted (p) vs true (d) values:
  - Mean Squared Error: $MSE = \langle (p_i - d_i)^2 \rangle$ — 0.0 is perfect
  - Correlation Coefficient: $r = \frac{\sum_{i=1}^{n} (p_i - \langle p \rangle)\,(d_i - \langle d \rangle)}{\sqrt{\sum_{i=1}^{n} (p_i - \langle p \rangle)^2}\,\sqrt{\sum_{i=1}^{n} (d_i - \langle d \rangle)^2}}$ — 1.0 is perfect
  - K-L Divergence: $KLD(D||P) = \sum_{k=1}^{bins} D_k \, ln \frac{D_k}{P_k}$ — 0.0 is perfect

Upstream Attached Region

- LotW predictions very narrow
- SMLP much better at predicting extreme values
- MSE and r improved for magnitude and components

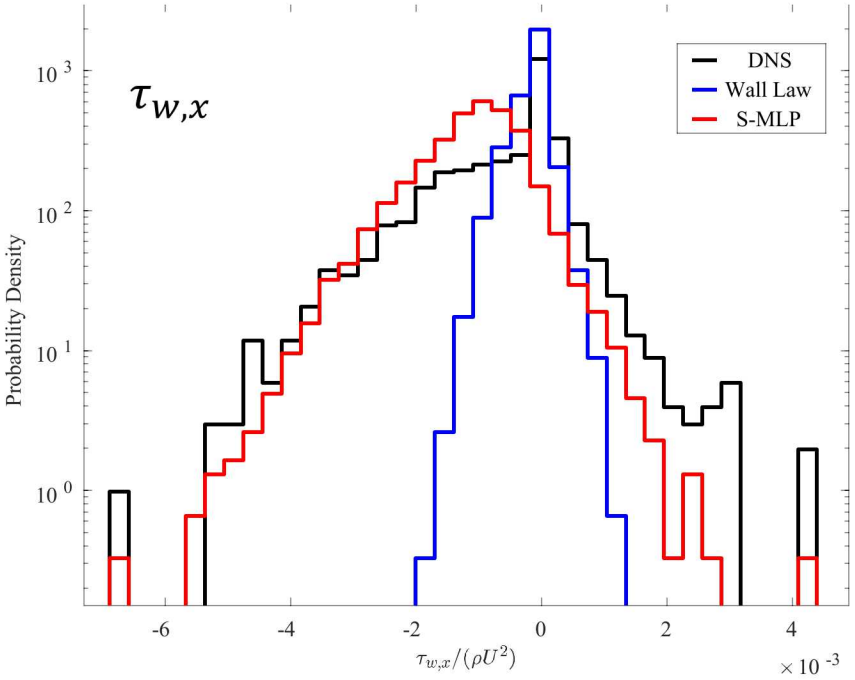


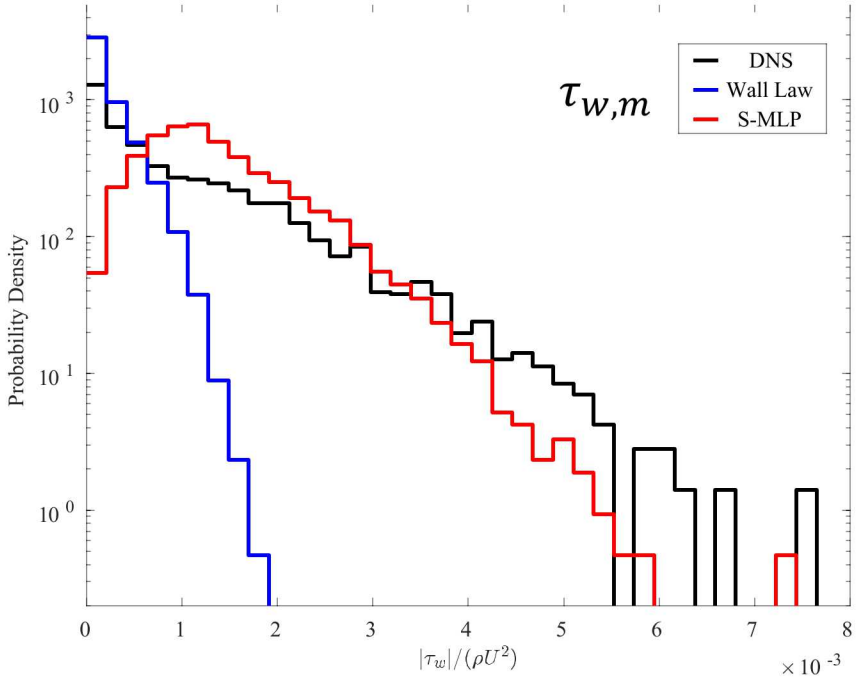


	MSE	r	KLD
SMLP $\tau_{w,m}$	9.9e-7	0.45	0.22
LotW $\tau_{w,m}$	1.1e-6	0.39	4.87
SMLP $\tau_{w,x}$	1.1e-6	0.45	0.22
LotW $\tau_{w,x}$	1.1e-6	0.39	4.89
SMLP $\tau_{w,z}$	1.6e-7	0.44	0.06
LotW $\tau_{w,z}$	2.3e-7	0.12	7.12

Downstream Detached Region

- LotW predictions very narrow
- SMLP much better at predicting extreme values
- SMLP misses τ_x zeros on step face
- MSE and r improved

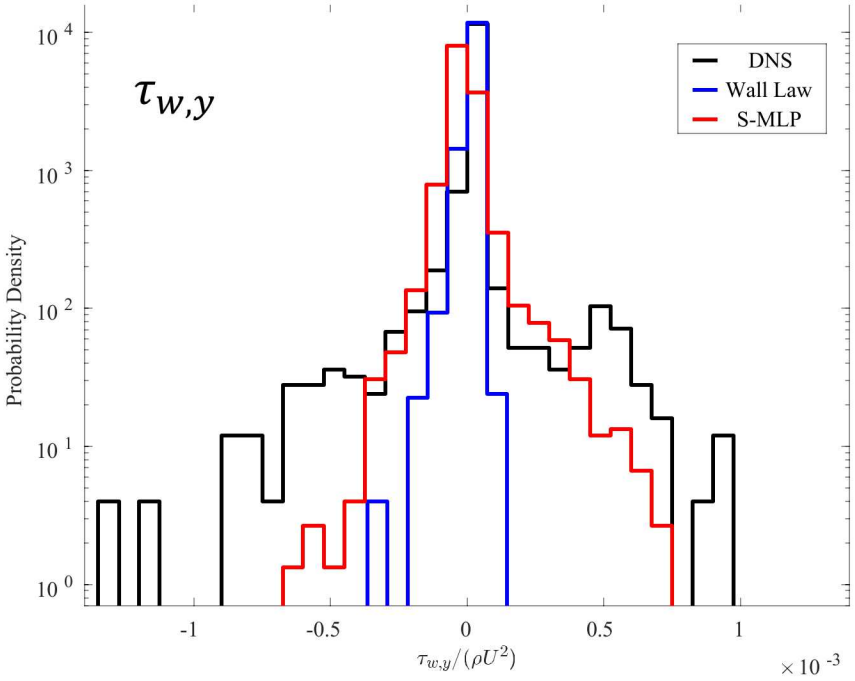


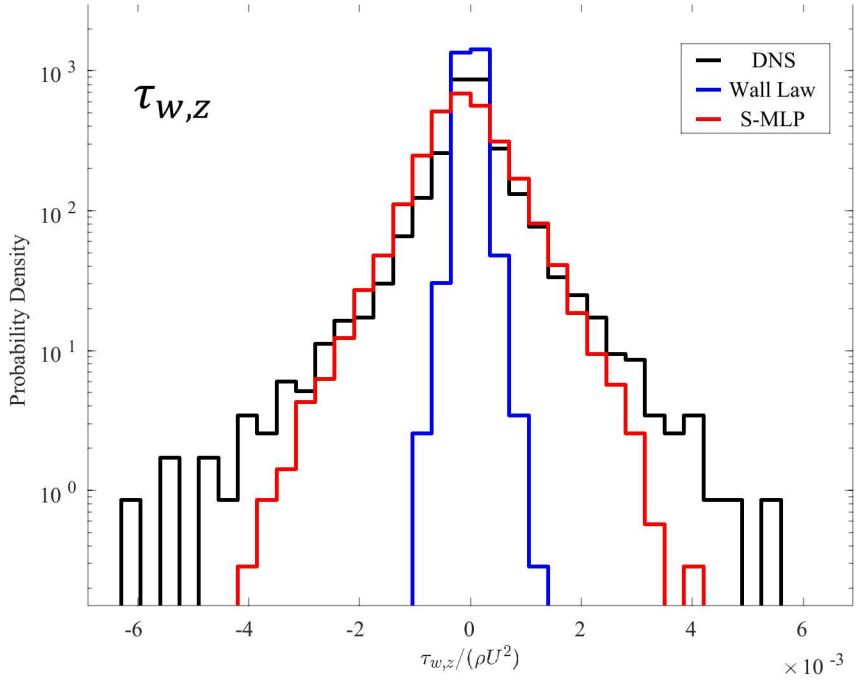


	MSE	r	KLD
SMLP $\tau_{w,m}$	8.3e-7	0.63	0.81
LotW $\tau_{w,m}$	1.5e-6	0.53	4.60
SMLP $\tau_{w,x}$	1.0e-6	0.64	0.74
LotW $\tau_{w,x}$	1.3e-6	0.57	3.13

Downstream Detached Region

- LotW predictions very narrow
- SMLP misses $\tau_{w,y}$ zeros on floor
- MSE and r both worse for $\tau_{w,z}$
  - Predicts extreme values but they are not that accurate





	MSE	r	KLD
SMLP $\tau_{w,y}$	9.3e-9	0.56	0.96
LotW $\tau_{w,y}$	6.0e-8	0.56	1.57
SMLP $\tau_{w,z}$	6.1e-7	0.48	0.16
LotW $\tau_{w,z}$	5.3e-7	0.51	2.69

Conclusions, Ongoing, & Future Work

- Invariant theory of $[\bar{\mathbf{S}}, \bar{\mathbf{\Omega}}, \bar{\mathbf{U}}, \vec{\mathbf{n}}]$ provides 8 Form-Invariants & 20 Scalar Invariants

$$\tau_{w,i} = \sum_{k=1}^{8} G^{(k)} \Pi_i^{(k)} \qquad G^{(k)} = \mathcal{F}^{(k)}(\lambda_1, \lambda_2, \lambda_3, \ldots \lambda_{20})$$

- Specialized architecture based on Siamese Multi-Layer Perceptron
- Upstream Attached Region:
  - SMLP better at predicting all components of $\tau_{w,i}$
- Downstream Detached Region:
  - SMLP better streamwise and vertical components
  - SMLP marginally worse at spanwise
  - SMLP misses zeros for wall-normal components on each face
    - Could be fixed by training, or *if*-statement in code implementation
  - Better in aggregate?

- Add cavity flow DNS data to training and testing..... ongoing
- Expand training dataset = More DNS frames..... ongoing
- Parametric study on hyperparameters to maximize SMLP accuracy..... future
- Test NN for Law of the Wall in CFD code (SIGMA-CFD, NaLU, SPARC)..... ongoing
- Implement SMLP for final, *a posteriori* model tests in LES code..... ongoing
- Paper at AIAA Aviation Forum, June 2019, Dallas, Texas

Thank you

Project funded under: Sandia's Laboratory Directed Research and Development (LDRD) program

A lot of other work was done with Neural Nets for pressure spectra and Random Forests for wall shear stress without invariant theory.

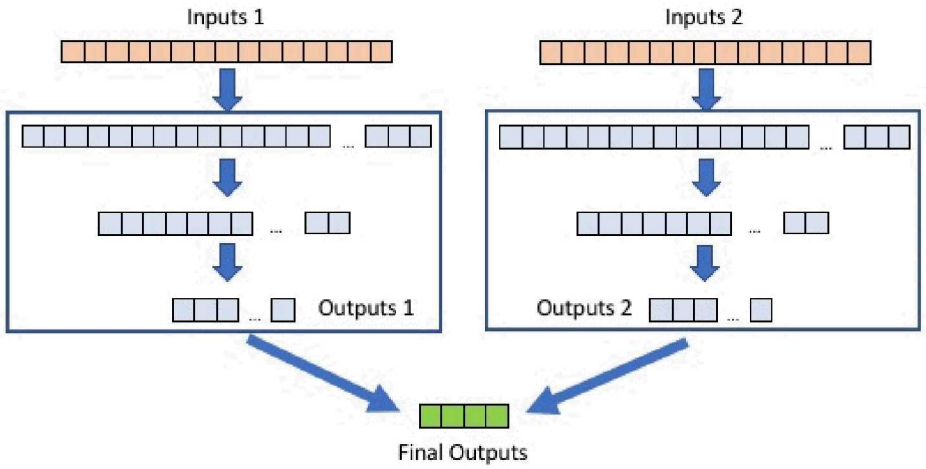
See Sandia Report: SAND2018-10602

Email: NMille1@Sandia.gov

Citations

1. 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.
2. 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.
3. 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.
4. 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.
5. 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.
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. Zheng, L., Duffner, S., Idrissi, K., Garcia, C., Baskurt, A., "Siamese Multi-layer Perceptrons for Dimensionality Reduction and Face Identification," *Multimedia Tools and Applications,* Vol. 75, 2016, pp. 5055–5073

Siamese Multi-Layer Perceptron



- Combination of open-source:
  Keras, TensorFlow
  and Sandia proprietary ML tools:
  TensorNet, Groupy
- Ensemble of 10 networks
- 4 hidden layers, ≈5100 weights
- ReLU operator
- Mean squared error loss function
- Logarithmic pre-scaling of all Invariants

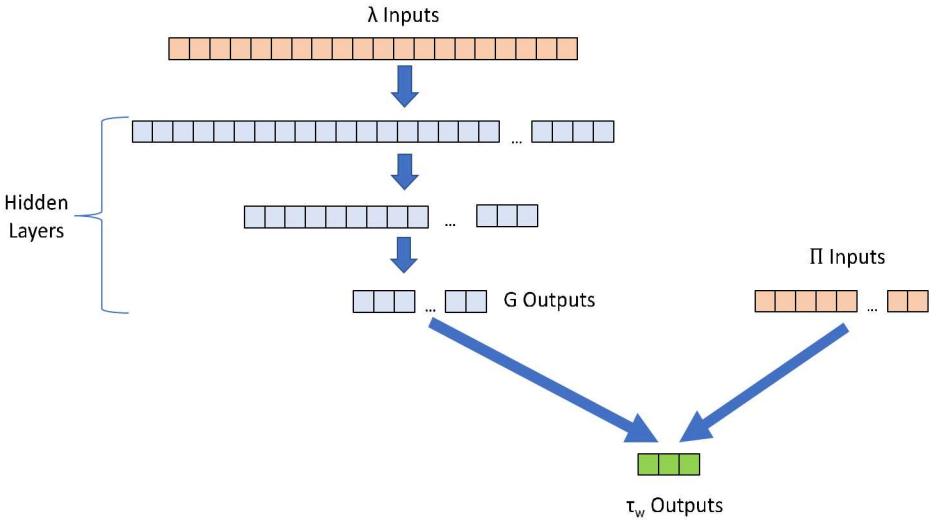
- 80/20 split on initial data for final testing
- 80/20 split on 80% for training/validation

Final layer operator combines outputs

- Minimize cosine similarity Zheng (2016)

- Matrix multiply for us:

$$\begin{bmatrix} \Pi_1^{(1)} & \Pi_1^{(2)} & \Pi_1^{(3)} & \dots & \Pi_1^{(8)} \\ \Pi_2^{(1)} & \Pi_2^{(2)} & \Pi_2^{(3)} & \dots & \Pi_2^{(8)} \\ \Pi_3^{(1)} & \Pi_3^{(2)} & \Pi_3^{(3)} & \dots & \Pi_3^{(8)} \end{bmatrix} \begin{bmatrix} G^{(1)} \\ G^{(2)} \\ G^{(3)} \\ \dots \\ G^{(8)} \end{bmatrix} = \begin{bmatrix} \tau_{w,x} \\ \tau_{w,y} \\ \tau_{w,z} \end{bmatrix}$$

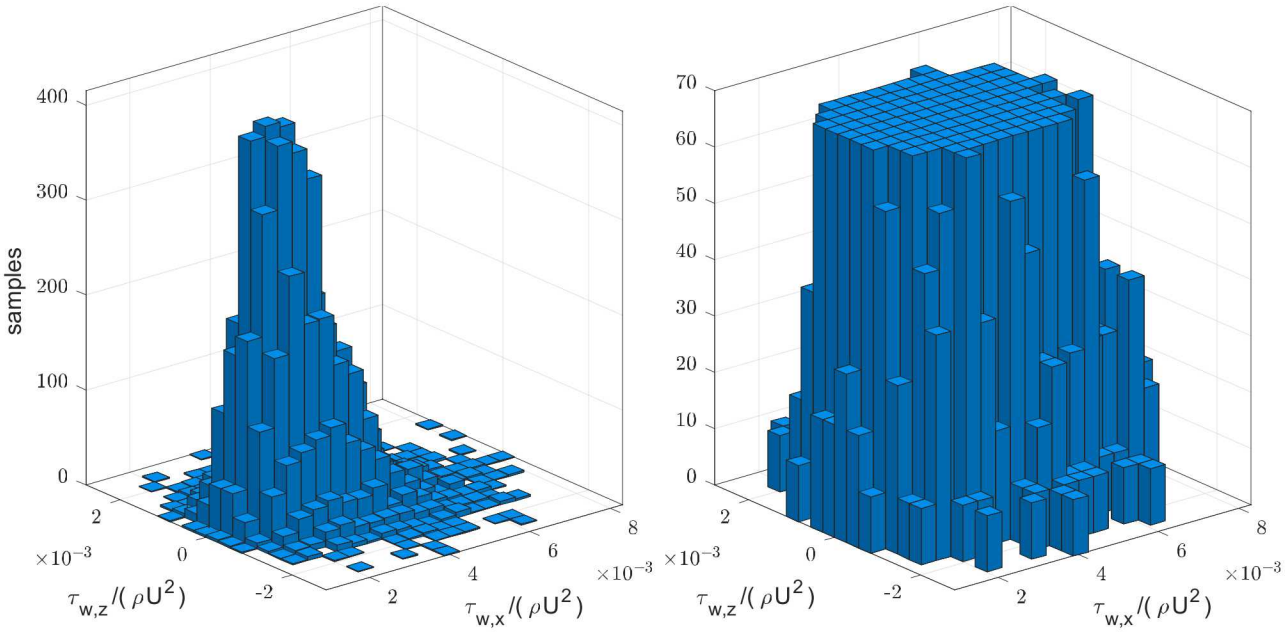


Pseudo-LES Filtering and Selective Sampling

- Upstream Boundary Layer Region:
  - 5800 pseudo-LES grid cells
  - $y^+ = [30,60,100,150]$
  - $[2\Delta_x^+, 2\Delta_z^+] = [80,50]$ & $[170,110]$ (Larsson *et al.* 2016)

- Downstream Detached Region:
  - 900 pseudo-LES grid cells on step face
  - 3660 pseudo-LES grid cells on "floor"
  - $y^+ = [15,30,60]$ from downstream attached BL
  - 8×8 DNS grid cells

- Data partitioned in 80% - 20% split
- 20% saved for final network testing
- 80% selectively sampled on nonzero stress components
- Further 80/20 split in training/validation



Backups: Quality Metrics

- Mean Squared Error:

$$MSE = \langle (p_i - d_i)^2 \rangle$$

- 0 is perfect

- Correlation Coefficient:

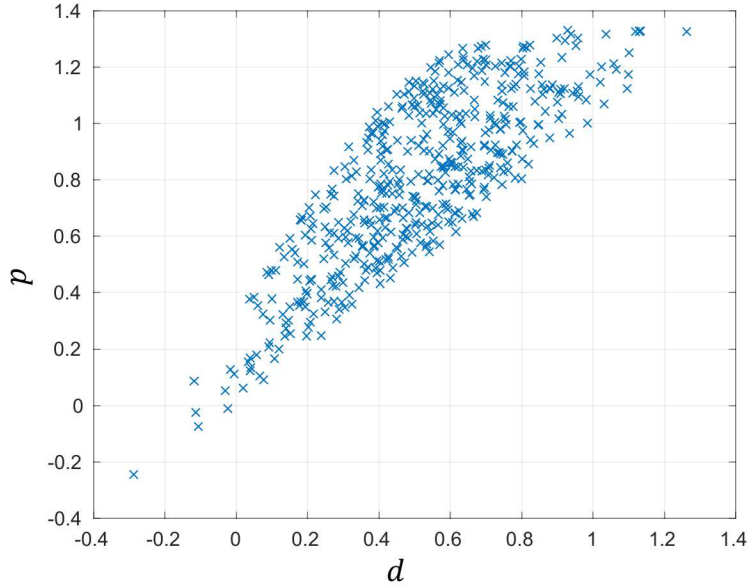
$$r = \frac{\sum_{i=1}^{n} (p_i - \langle p \rangle)\,(d_i - \langle d \rangle)}{\sqrt{\sum_{i=1}^{n} (p_i - \langle p \rangle)^2}\,\sqrt{\sum_{i=1}^{n} (d_i - \langle d \rangle)^2}}$$

- 1.0 is perfect

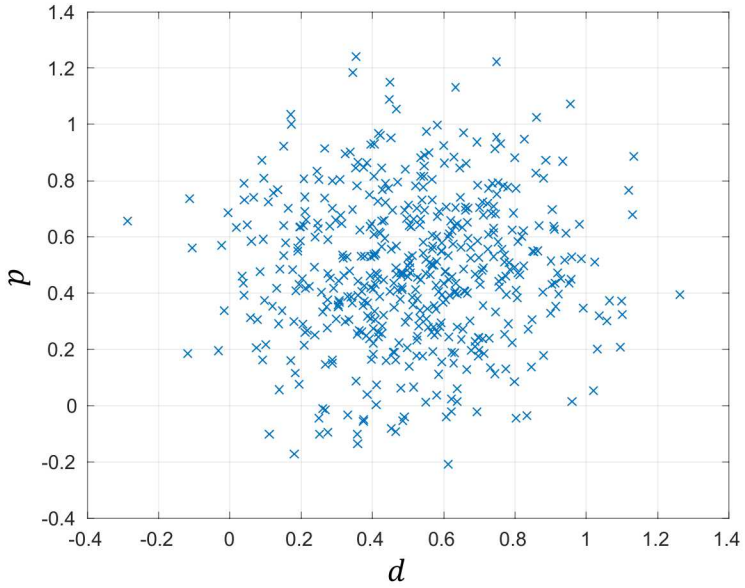
- K-L Divergence:

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

- 0 is perfect

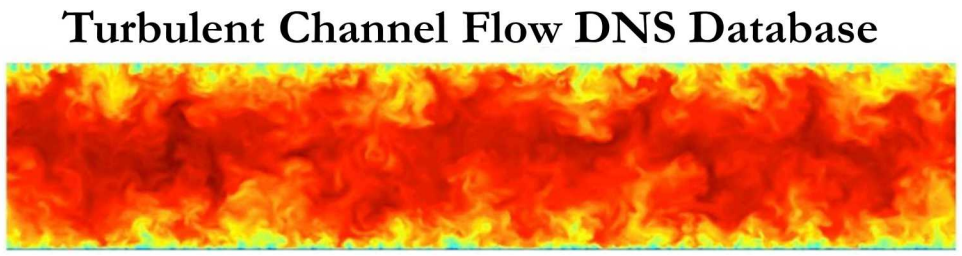


MSE = 0.13, r = 0.78, KLD = 0.66



MSE = 0.13, r = 0.03, KLD = 0.04

Other NN studies done:



- Stencil of 45 LES grid cells above point of interest on the wall
- Variety of state variables tested
  - Random Forests used for feature importance ranking

- In MLPs:
  - Tested accuracy vs. number of training vectors
    - More vectors & more features reduced error, but had limiting returns
  - Tested usefulness of pre-sampling strategies
    - Pre-sampling produced much better training results

No. Training Pts.	MSE	K-L Div.	$r(\tilde{\tau}_p, \tilde{\tau}_d)$	$\epsilon_{<\tilde{\tau}>}$, %	$\epsilon_{\tilde{\tau}'}$, %
67K	8.77e-7	0.141	0.532	16.5	16.4
126K	7.76e-7	0.131	0.537	14.5	14.0
181K	7.13e-7	0.121	0.543	13.1	12.3
373K	7.17e-7	0.117	0.567	14.1	13.5
110K, rand. samp.	3.43e-7	1.03	0.629	0.5	-2.2
τ_{log}	5.97e-7	1.17	0.482	-13.1	-15.6