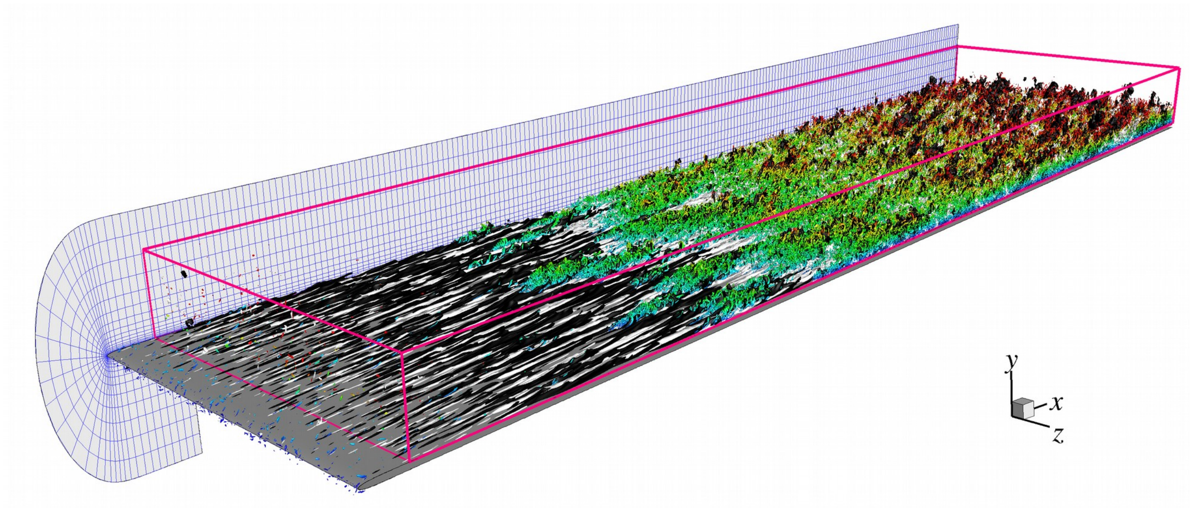
ML, boundary layer transition, and Reynolds stresses

Shaun Harris (srharris@stanford.edu)
Mechanical Engineering

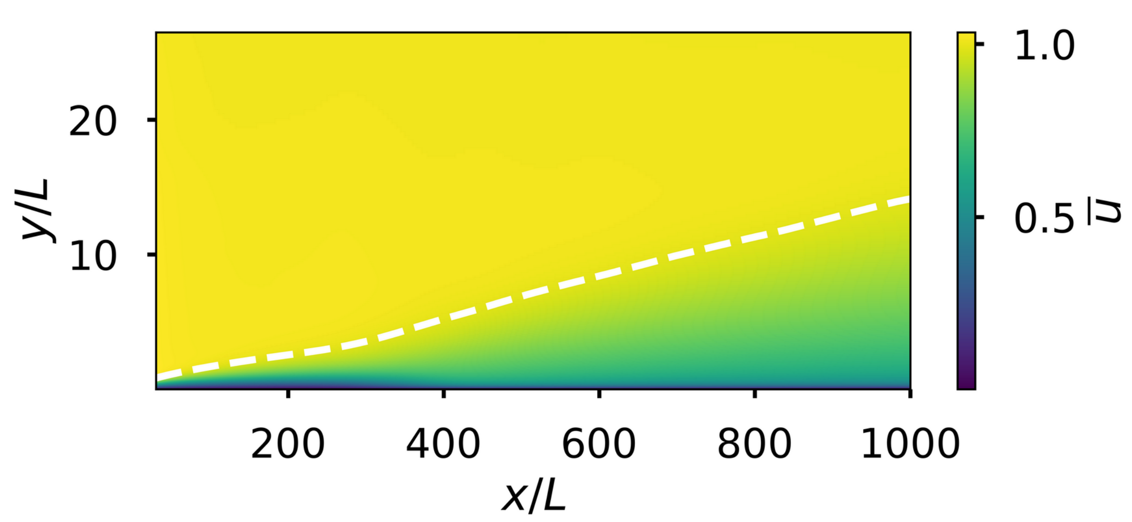
Stanford
CS229: Project

Motivation and Data

Computation fluid dynamics (CFD) relies on computing the nonlinear perturbation terms that arise in the Navier-Stokes equations.
I explore various techniques to model these terms using ML given the average base flow gradients and compare the models for a transitional boundary layer.
Data obtained from the JHTDB at http://turbulence.pha.jhu.edu



White and black: $u' = \pm 0.1 U_\infty$
Color: $\lambda_2 = -0.1 U_\infty^2 / L^2$

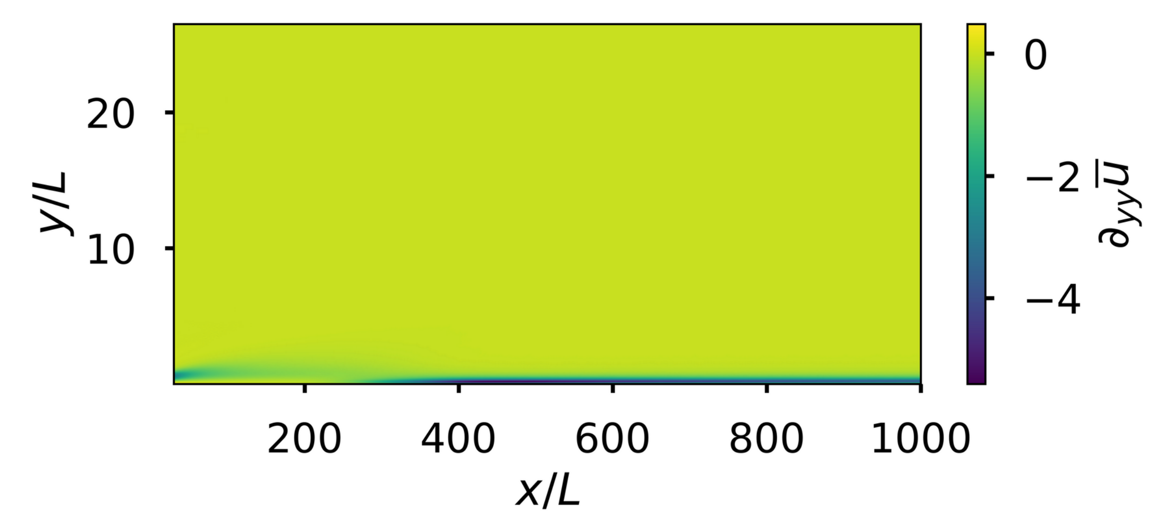


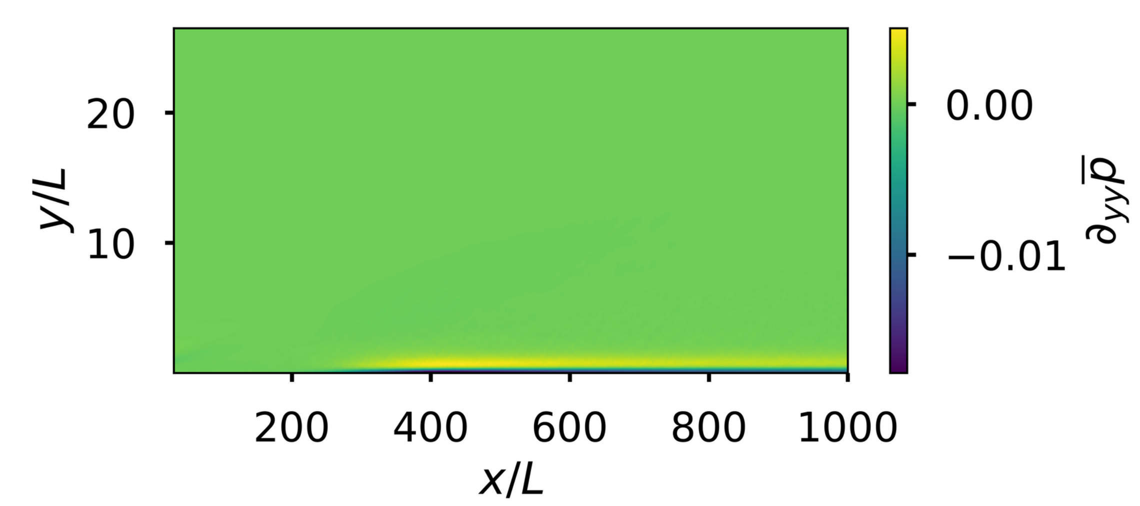
Streamwise average base flow

Features

To keep the input features Galilean invariant and have physical meaning, the **input features** selected included averaged velocity gradients, pressure, pressure gradients and kinematic viscosity:

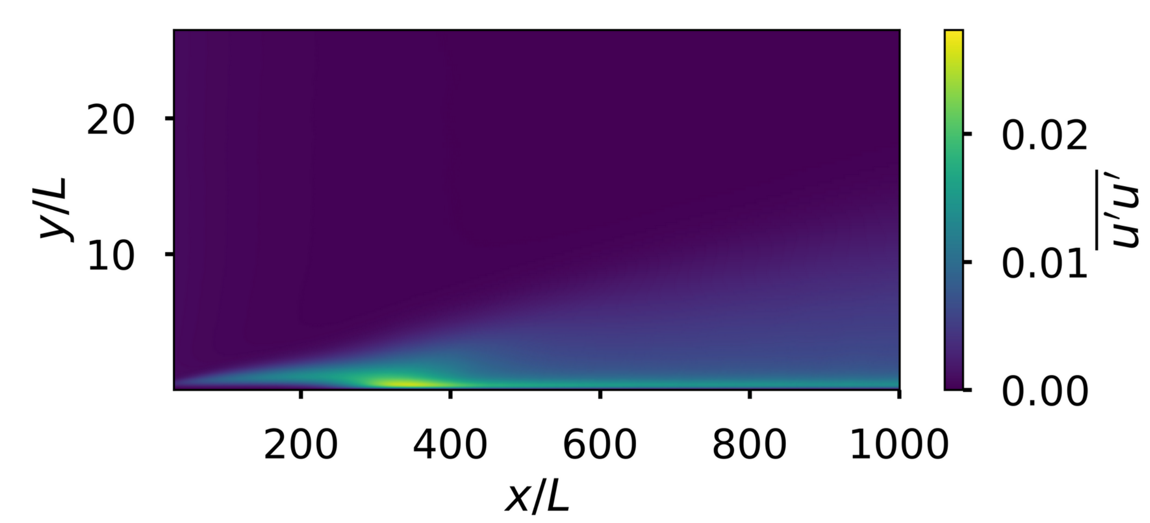
$$\overline{P} \quad \nabla\vec{\overline{u}} \quad \nabla\overline{P} \quad \nabla^2\vec{\overline{u}} \quad \nabla^2\overline{P} \quad \nu$$





The **output features** included the nonlinear averaged perturbation terms:

$$\overline{u'u'} \quad \overline{v'v'} \quad \overline{w'w'} \quad \overline{u'v'}$$



Models

The models tested here include **linear regression (LR)** and a simple **neural network (NN)** (two 100 neuron layers with ReLU activation).
Data augmentation included adding random values to the input features on the order of 2% and by generating polynomial and interaction features of degree 2 and 3.

Four models tested and shown here:
1)LR without aug.
2)LR with aug. poly2
3)LR with aug. poly3
4)NN without aug.

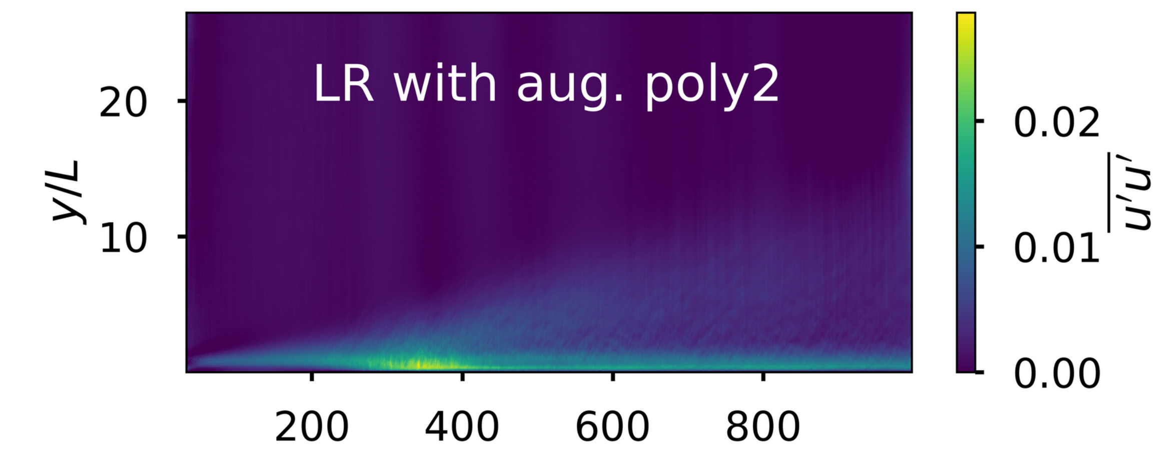
R^2 score was used to compare the resulting models.

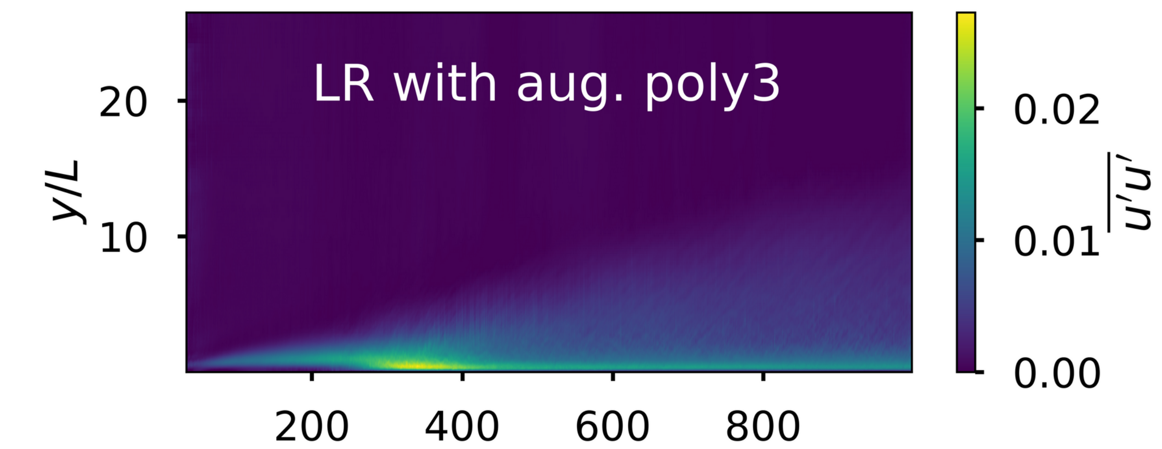
$$R^2 \equiv 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$$

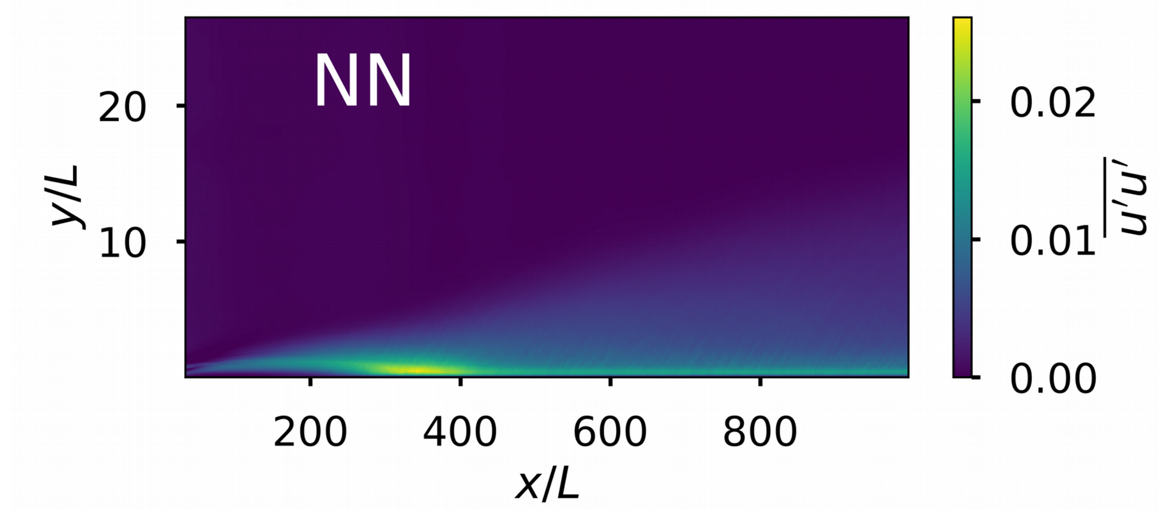
R^2	Training	Test
1) LR	0.728279	0.729059
2) LR poly2	0.951162	0.952244
3) LR poly3	0.992689	0.993555
4) NN	0.990437	0.990409

scikit learn PyTorch ELI5

Results and Discussion







- Linear regression with polynomial feature expansion of degree 3 yielded the best fit on a 70/30 data split validation.
- A simple neural network followed closely behind.
- Permutation Importance or Mean Decrease Accuracy on LR showed the most important input features to be for poly2: $\frac{\partial^2 \overline{u}}{\partial y^2}$ $\frac{\partial^2 \overline{P}}{\partial y^2}$

  and for poly3: $\frac{\partial \overline{u}}{\partial x}$ $\frac{\partial \overline{v}}{\partial y}$