Change of Basis of Fluid Flow Data as a Method to Improve Convergence when Tuning Turbulence Models with Machine Learning

Final project in IMS135

Pontus Nilsson & Viktor Sundström

October 21, 2024

Background

- RANS-equations need turbulence model
- k-ε contains 6 constants
  - 2 when simplified (channel flow)
- (ML) Functions instead of constants?

$$\frac{\partial \bar{v}_i}{\partial x_i} = 0$$

$$\frac{\partial \rho_0 \bar{v}_i}{\partial t} + \frac{\partial}{\partial x_j}(\rho_0 \bar{v}_i \bar{v}_j) = -\frac{\partial \bar{p}}{\partial x_i} + \mu \frac{\partial^2 \bar{v}_i}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} - \beta \rho_0 (\bar{\theta} - \theta_0) g_i$$

$$\begin{aligned}
\overline{v_1'^2} &= \frac{k}{12}\tau^2 \left(\frac{\partial \bar{v}_1}{\partial x_2}\right)^2 (c_0 + 6c_2) + \frac{2}{3}k \\
\overline{v_2'^2} &= \frac{k}{12}\tau^2 \left(\frac{\partial \bar{v}_1}{\partial x_2}\right)^2 (c_0 - 6c_2) + \frac{2}{3}k \\
\overline{v_3'^2} &= -\frac{k}{6}\tau^2 \left(\frac{\partial \bar{v}_1}{\partial x_2}\right)^2 c_0 + \frac{2}{3}k \\
\overline{v_1' v_2'} &= -c_\mu \tau \frac{\partial \bar{v}_1}{\partial x_2}
\end{aligned}$$

Limitations / Scope

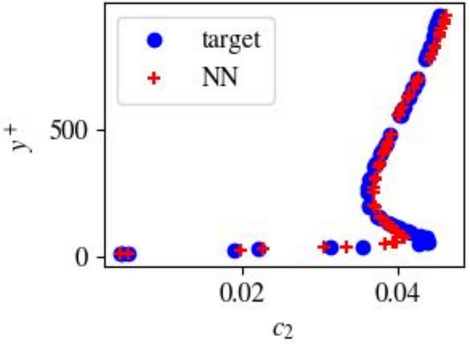
- Fixed Network architecture / size / training

$$2 \rightarrow 50 \rightarrow 50 \rightarrow 50 \rightarrow 25 \rightarrow 2$$

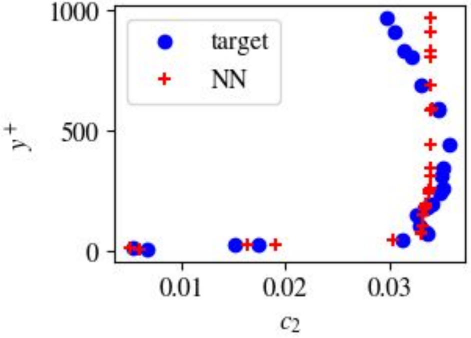
- 2 variables
- DNS datasets: Boundary layer and Channel flow.
- $5 < y^+ < 1000$

Validation against different dataset

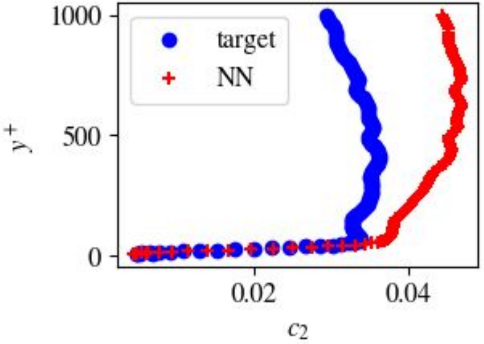
Channel flow target,
Channel flow training



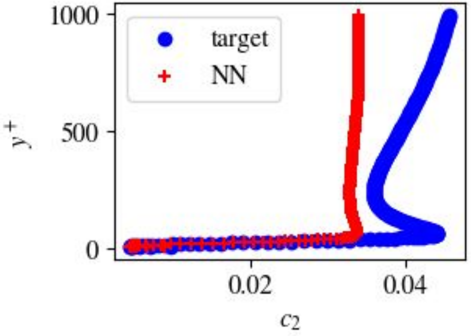
Boundary layer target,
Boundary layer training
(Might have needed
more training)



Boundary layer target,
Channel flow training

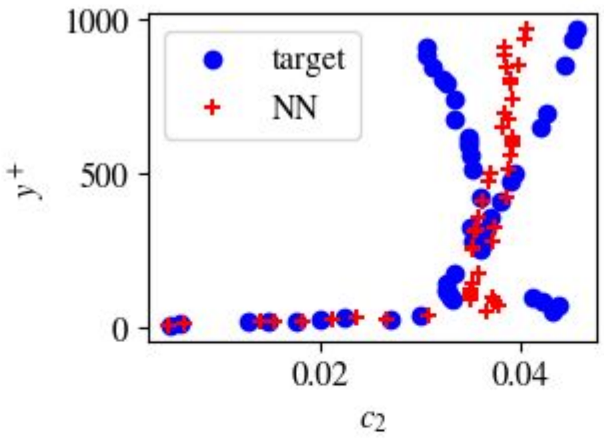


Channel flow target,
Boundary layer training

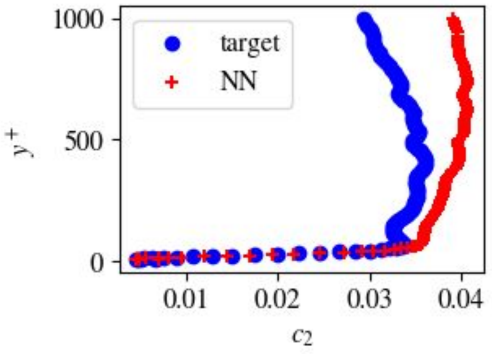


Note that the neural network predicts c_2 to be of a similar size to what it was trained on

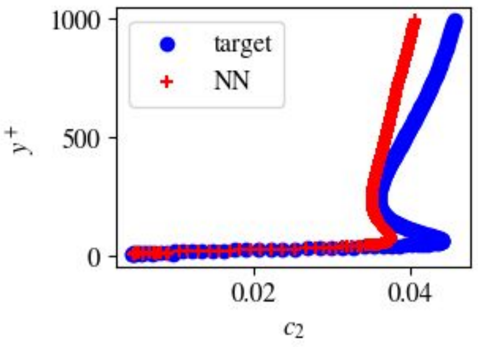
Training on both datasets



Boundary layer target



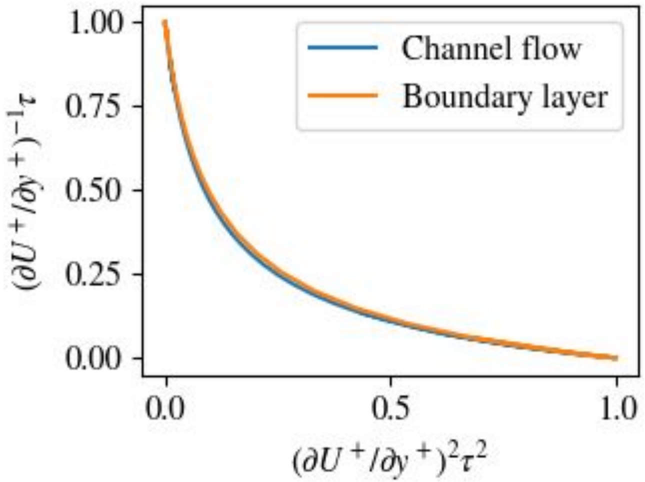
Channel flow target

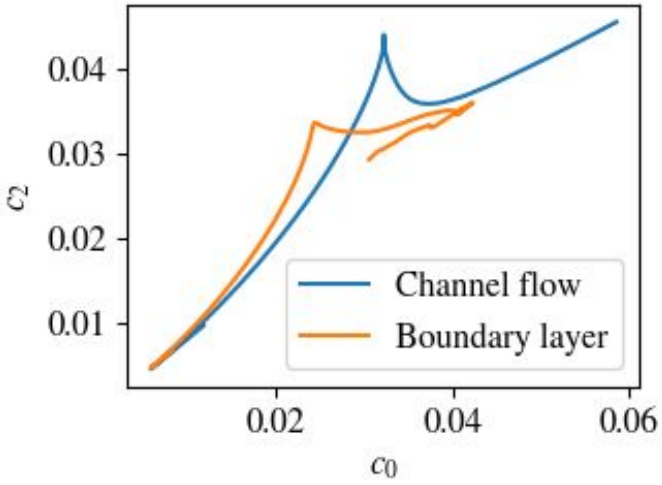


A neural network is a function

f(x1,x2) → (y1,y2)

Same (x1,x2) will always return same (y1,y2)!





X1 and X2?

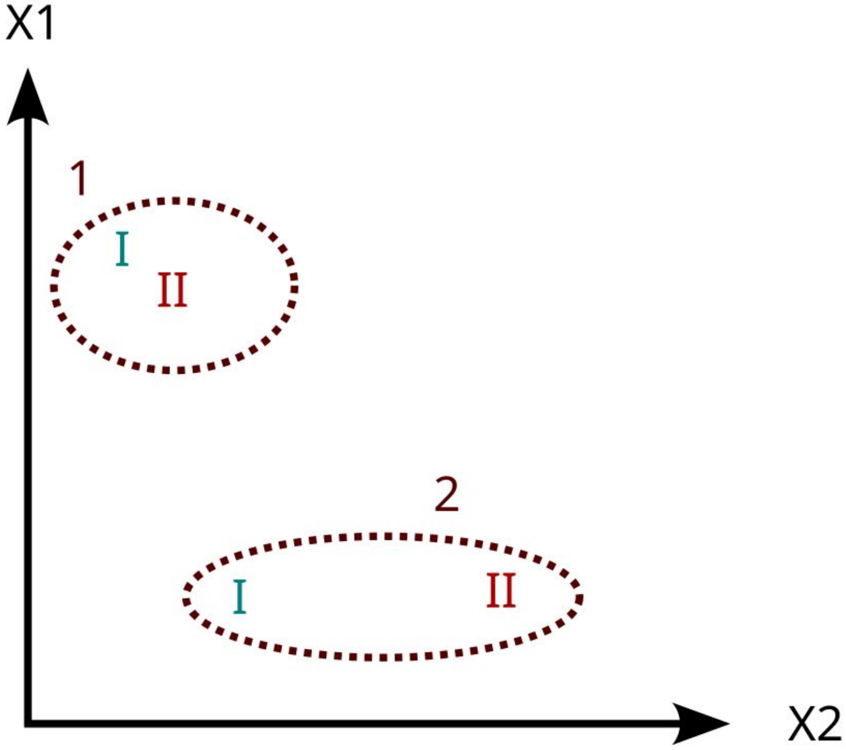
$$y^+, k^+, \epsilon^+, U^+, \frac{\partial U^+}{\partial y^+}$$

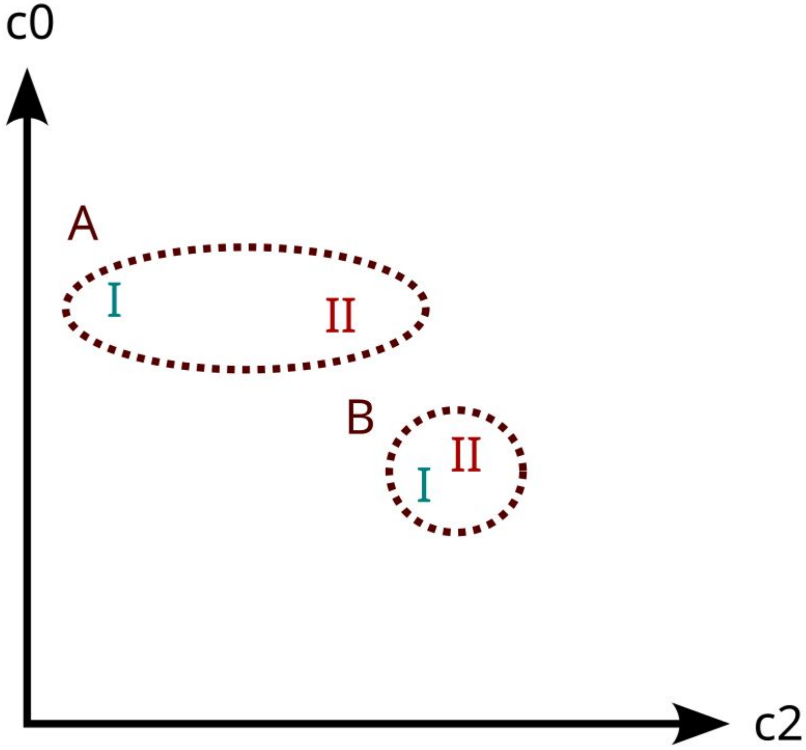
$$\left(\partial U^+/\partial y^+\right)^2 \quad , \quad \left(\partial U^+/\partial y^+\right)^{-1}$$

$$T^2\left(\partial U/\partial y\right)^2 \quad , \quad T\left(\partial U/\partial y\right)^{-1}, \quad T = k/\varepsilon$$

$$\left(\partial U^+/\partial y^+\right)^2 \quad , \quad k^+/\varepsilon^+$$

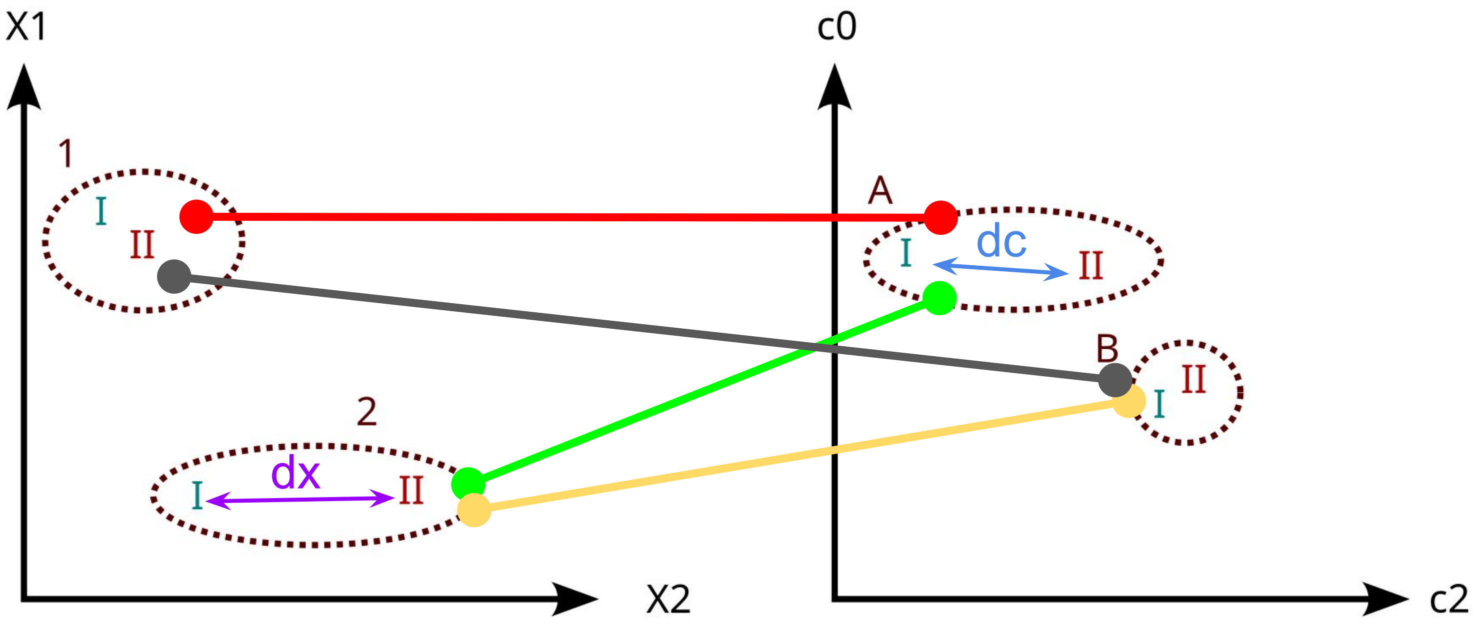
A good choice of X1 and X2?





How to separate datasets

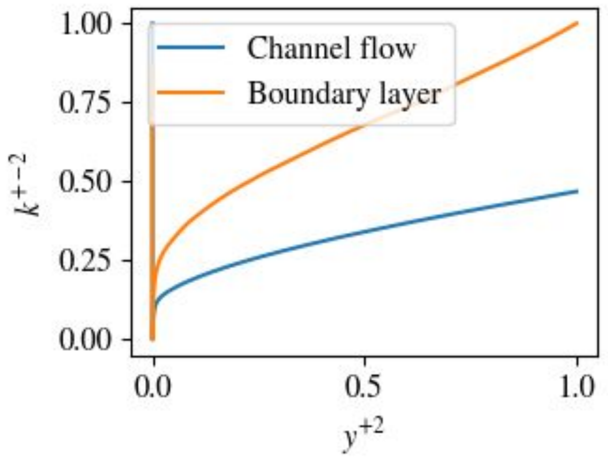
$$dd = \sum_{\text{closest pairs}} \left(dx\,(\,\mathrm{I}\,,\,\mathrm{II}\,)^2 \times (\,dc\,(\,\mathrm{I}\,,\,\mathrm{II}\,))\,\right)$$



Results

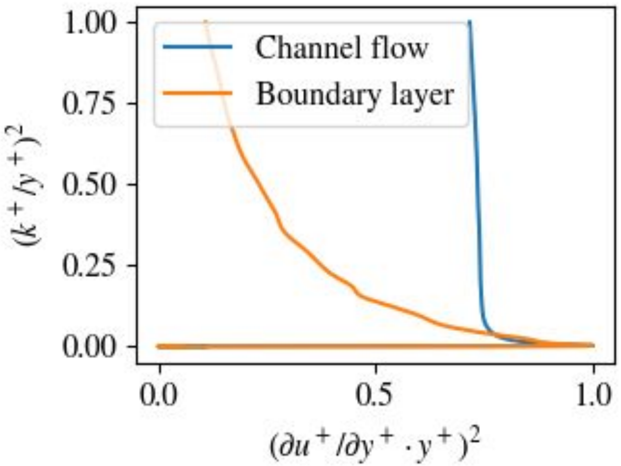
$$\left(y^+\right)^2 \Big/ \left(k^+\right)^{-2}$$

dd = 1.4E-03



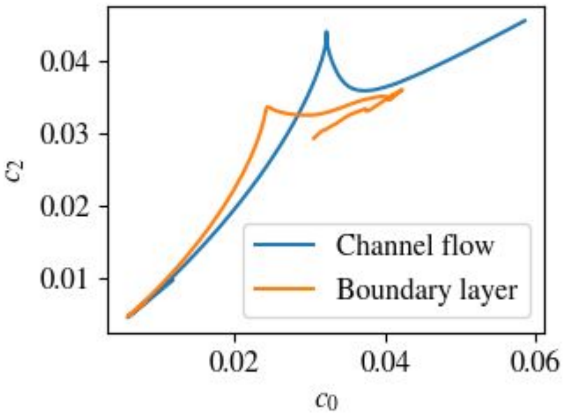
$$\left(y^+ \times \frac{\partial U^+}{\partial y^+}\right)^2 \Big/ \left(\frac{k^+}{y^+}\right)^2$$

dd = 3.0E-03



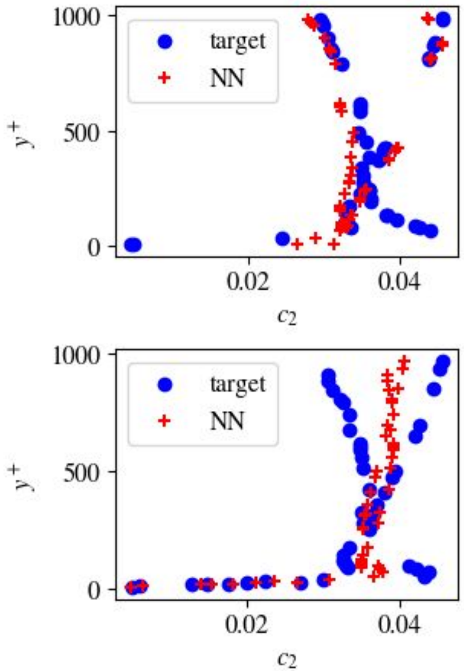
Original

dd = 4.8E-05

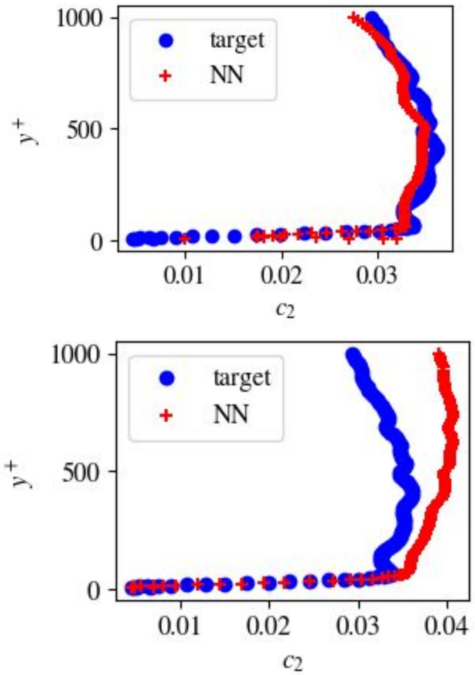


$(k^+)^{-2}$ and $(y^+)^2$

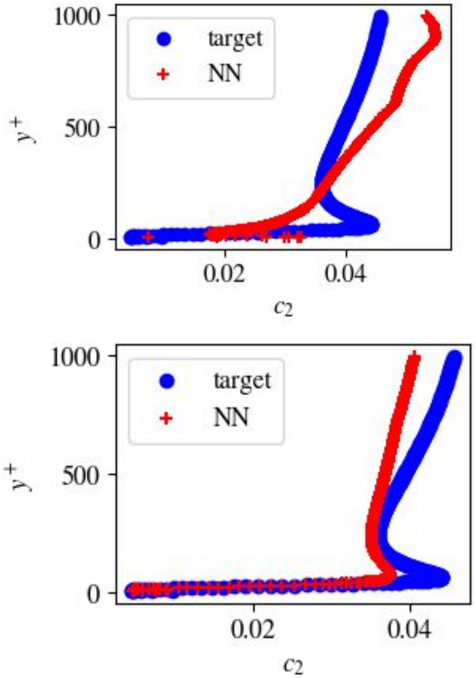
Training on both



Boundary layer



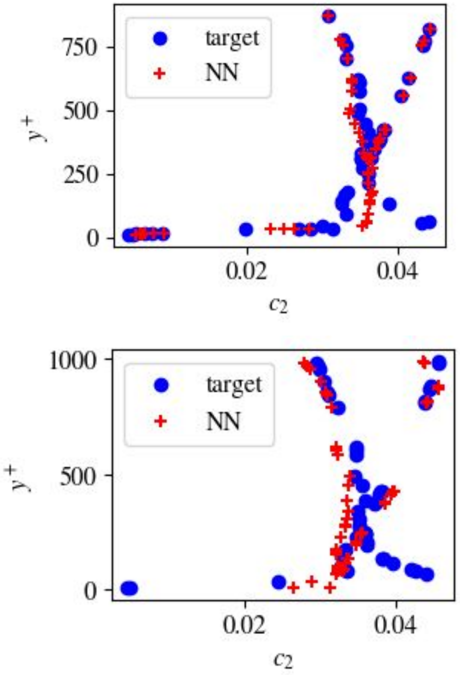
Channel flow



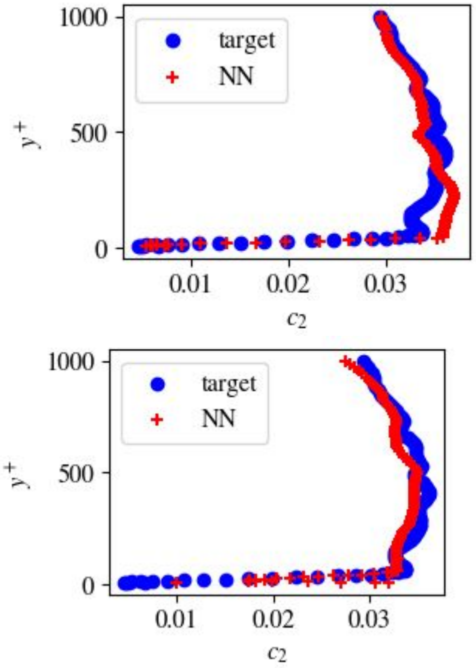
Comparison with original choice for x

$(du^+/dy^+ * y^+)^{-2}$ and $(k^+/y^+)^2$

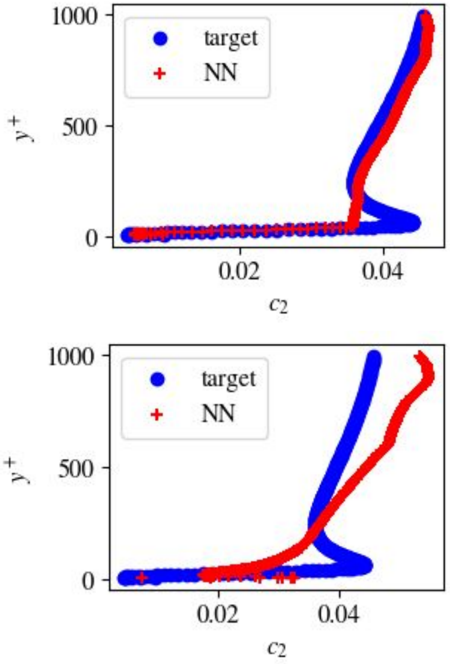
Training on both



Boundary layer



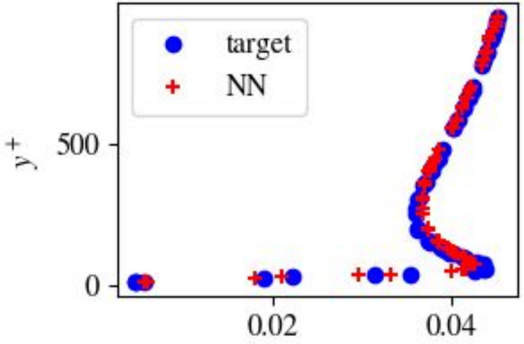
Channel flow



Comparison with k^+ and y^+

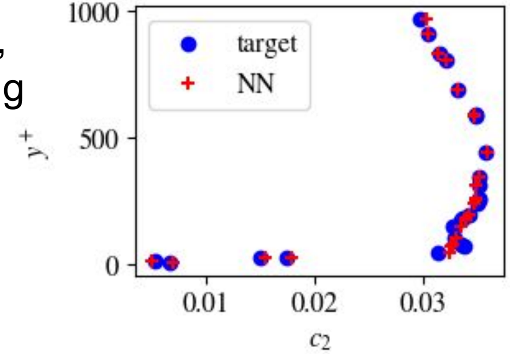
Only training on one dataset

Channel flow target,
Channel flow training

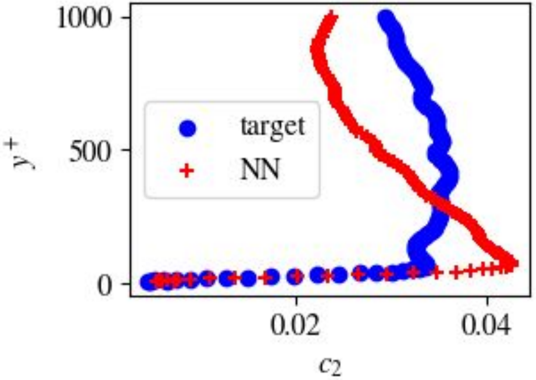


Boundary layer target,
Boundary layer training

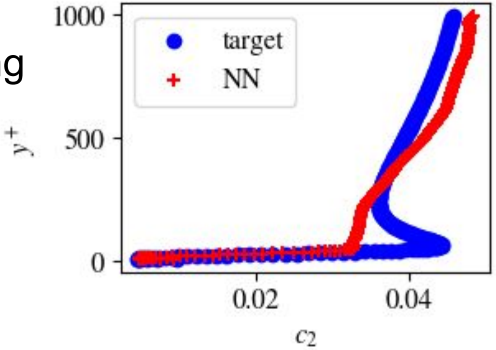
Might need more training



Boundary layer target,
Channel flow training



Channel flow target,
Boundary layer training



Note that the lack of training for the validation dataset still leads to a not that good fit

Conclusion

- We could train on more sets of data
- By changing variables Network performances increases
- Choice of variable found by defining metric
  - Quick computation on raw datasets

References

- Davidson L., NN-train-BL.py,

  https://www.tfd.chalmers.se/~lada/ML-IMS135/

- Project Github

  https://github.com/vrogly/machine-learning-for-turbulence

Thank you for listening!

Appendix

16081	10708	('dudy^-1_u^2', 'k^-2_eps^1')	dudy	-1	u	2	k	-2	eps	1	0.032	0.013	8.0E-06
16082	6612	('dudy^-1_k^-1', 'k^-2_eps^2')	dudy	-1	k	-1	k	-2	eps	2	0.031	0.013	7.9E-06
16083	10709	('dudy^-1_u^2', 'k^-2_eps^2')	dudy	-1	u	2	k	-2	eps	2	0.031	0.013	7.7E-06
16084	12485	('yplus^1_k^1', 'yplus^1_u^-1')	yplus	1	k	1	yplus	1	u	-1	0.018	0.013	7.7E-06
16085	12498	('yplus^1_k^1', 'k^-1_eps^1')	yplus	1	k	1	k	-1	eps	1	0.027	0.013	7.7E-06
16086	6552	('dudy^-1_k^-1', 'dudy^-1_u^2')	dudy	-1	k	-1	dudy	-1	u	2	0.020	0.014	7.7E-06
16087	10605	('dudy^-1_u^1', 'k^-2_eps^2')	dudy	-1	u	1	k	-2	eps	2	0.032	0.013	7.7E-06
16088	14633	('yplus^1_u^-1', 'k^-2_eps^1')	yplus	1	u	-1	k	-2	eps	1	0.030	0.013	7.7E-06
16089	14583	('yplus^1_u^-2', 'k^-1_eps^1')	yplus	1	u	-2	k	-1	eps	1	0.027	0.013	7.6E-06
16090	12582	('yplus^1_k^2', 'k^-1_eps^1')	yplus	1	k	2	k	-1	eps	1	0.029	0.012	7.6E-06
16091	14580	('yplus^1_u^-2', 'k^-2_eps^2')	yplus	1	u	-2	k	-2	eps	2	0.031	0.013	7.5E-06
16092	12495	('yplus^1_k^1', 'k^-2_eps^2')	yplus	1	k	1	k	-2	eps	2	0.031	0.013	7.5E-06
16093	10080	('dudy^-2_u^-1', 'k^2_eps^-1')	dudy	-2	u	-1	k	2	eps	-1	0.021	0.016	7.5E-06
16094	9898	('dudy^-2_u^-2', 'dudy^-1_u^-1')	dudy	-2	u	-2	dudy	-1	u	-1	0.020	0.014	7.4E-06
16095	6205	('dudy^-2_k^1', 'k^2_eps^-1')	dudy	-2	k	1	k	2	eps	-1	0.021	0.016	7.3E-06
16096	15396	('k^1_eps^-1', 'eps^-1_u^-1')	k	1	eps	-1	eps	-1	u	-1	0.020	0.015	7.3E-06
16097	14579	('yplus^1_u^-2', 'k^-2_eps^1')	yplus	1	u	-2	k	-2	eps	1	0.031	0.013	7.3E-06
16098	12494	('yplus^1_k^1', 'k^-2_eps^1')	yplus	1	k	1	k	-2	eps	1	0.031	0.013	7.1E-06
16099	3205	('u^1_0^0', 'k^-1_u^2')	u	1	0	0	k	-1	u	2	0.021	0.014	7.1E-06
16100	3204	('u^1_0^0', 'k^-1_u^1')	u	1	0	0	k	-1	u	1	0.024	0.014	7.1E-06
16101	3069	('u^1_0^0', 'u^2_0^0')	u	1	0	0	u	2	0	0	0.021	0.014	7.1E-06
16102	0	('dudy^-2_0^0', 'dudy^-1_0^0')	dudy	-2	0	0	dudy	-1	0	0	0.018	0.014	7.0E-06
16103	3364	('u^2_0^0', 'k^-1_u^1')	u	2	0	0	k	-1	u	1	0.021	0.014	6.9E-06
16104	15371	('k^1_eps^-1', 'k^2_eps^-2')	k	1	eps	-1	k	2	eps	-2	0.017	0.015	6.8E-06
16105	3365	('u^2_0^0', 'k^-1_u^2')	u	2	0	0	k	-1	u	2	0.019	0.014	6.5E-06
16106	10715	('dudy^-1_u^2', 'k^1_eps^-1')	dudy	-1	u	2	k	1	eps	-1	0.021	0.015	6.3E-06
16107	12484	('yplus^1_k^1', 'yplus^1_u^-2')	yplus	1	k	1	yplus	1	u	-2	0.019	0.013	6.3E-06
16108	10611	('dudy^-1_u^1', 'k^1_eps^-1')	dudy	-1	u	1	k	1	eps	-1	0.019	0.014	6.2E-06
16109	15395	('k^1_eps^-1', 'eps^-1_u^-2')	k	1	eps	-1	eps	-1	u	-2	0.024	0.014	6.2E-06
16110	6618	('dudy^-1_k^-1', 'k^1_eps^-1')	dudy	-1	k	-1	k	1	eps	-1	0.018	0.014	6.0E-06
16111	3394	('dudy^-2_yplus^-2', 'dudy^-1_yplus^-1')	dudy	-2	yplus	-2	dudy	-1	yplus	-1	0.024	0.012	5.4E-06