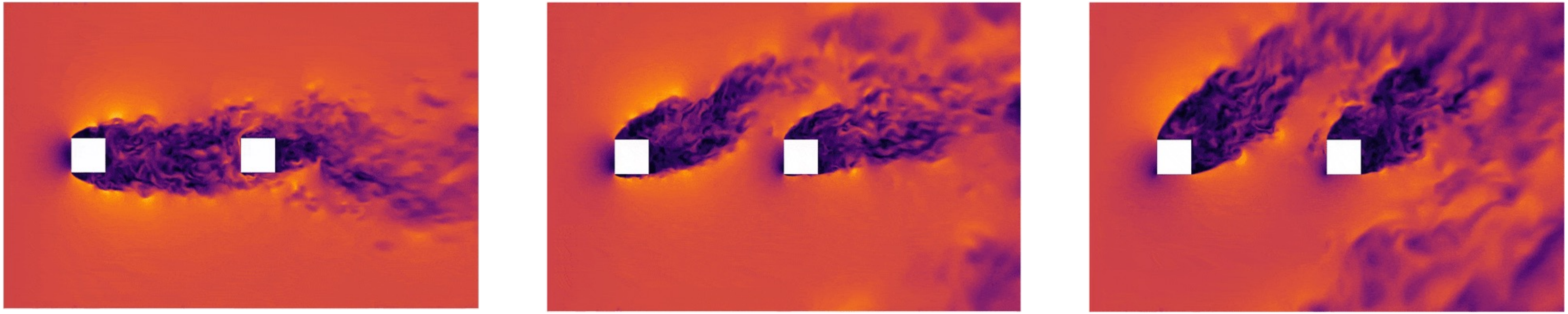


XGBoost-augmented RANS closure modelling of complex 3D flows

4/11/23

Ryley McConkey*, **Andrew Mole^**, Alex Skillen^, Alistair Revell^, Eugene Yee*, and Fue-Sang Lien*

* University of Waterloo, Canada

^ University of Manchester, UK

UNIVERSITY OF WATERLOO | FACULTY OF ENGINEERING

Introduction

- NASA CFD 2030 plan demonstrates need for accurate models for ***industrial*** flows (Slotnick et al. 2014).
- ***RANS*** still not as accurate as required for industry.
- Accurate methods (***LES***) exist but will be ***expensive*** for decades (Widtherden 2016).
- **How can we improve RANS in the meantime?**

Slotnick et al. 2014, Vincent et al. 2016

UNIVERSITY OF WATERLOO | FACULTY OF ENGINEERING

Introduction

Corrective, Data-driven RANS closure models

Main idea: Run a RANS simulation, correct the Reynolds stress tensor, then re-run simulation

Machine learning: Trains a model to predict a "better" Reynolds stress tensor (e.g., from DNS or LES) from a RANS simulation

Key issues:

- Training dataset
- Machine learning model architecture & input features
- Conditioning & injection

Introduction

Novelties:

- Explore applicability of machine learning techniques for an industrially relevant, ***complex, 3D flow***.
  - Previous studies have focused on ***canonical 2D*** flows (Ling et al. 2016, Kaandorp & Dwight 2020, McConkey et al. 2022)

- Explore stability and robustness of injected ***XGBoost*** predictions
  - Previous studies have used random forests (Kaandorp & Dwight 2020), and neural networks (Ling et al. 2016, McConkey et al. 2022)

Methodology

What should we ***predict*** with the model?

Model predicts: $\nu_t^\dagger$, $a^\perp$

$$\nabla \cdot (\vec{U}\vec{U}) = -\nabla p + \nu \nabla^2 \vec{U} - \nabla \cdot \tau$$

$$a \equiv \tau - \frac{1}{3}\mathrm{tr}(\tau) I$$

$$a = -2\nu_t^\dagger S + a^\perp$$

"Optimal eddy viscosity"

$$\nu_t^\dagger = \arg\min_{\nu_t \geq 0} ||a - (-2\nu_t S)||$$

"Remainder"

$$a^\perp = a - (-2\nu_t^\dagger S)$$

:ULTY OF
:INEERING

Methodology

What ***input features*** should we use?

Key fields:
U, p, k

Tensors:
S, R, Ap, Ak

$$\begin{bmatrix} 0 & -\partial_z k & \partial_y k \\ \partial_z k & 0 & -\partial_x k \\ -\partial_y k & \partial_x k & 0 \end{bmatrix}$$

Form a **minimal integrity basis** for these 4 tensors

Extract the maximum ***invariant information*** from these fields.

47 basis tensors

Take **tensor invariants**

$$I_1(A) = \mathrm{tr}(A)$$
$$I_2(A) = \frac{1}{2}[(\mathrm{tr}(A))^2 - \mathrm{tr}(A^2)]$$

94 invariant scalars

Input features

UNIVERSITY OF WATERLOO | FACULTY OF ENGINEERING

Methodology

Input Features

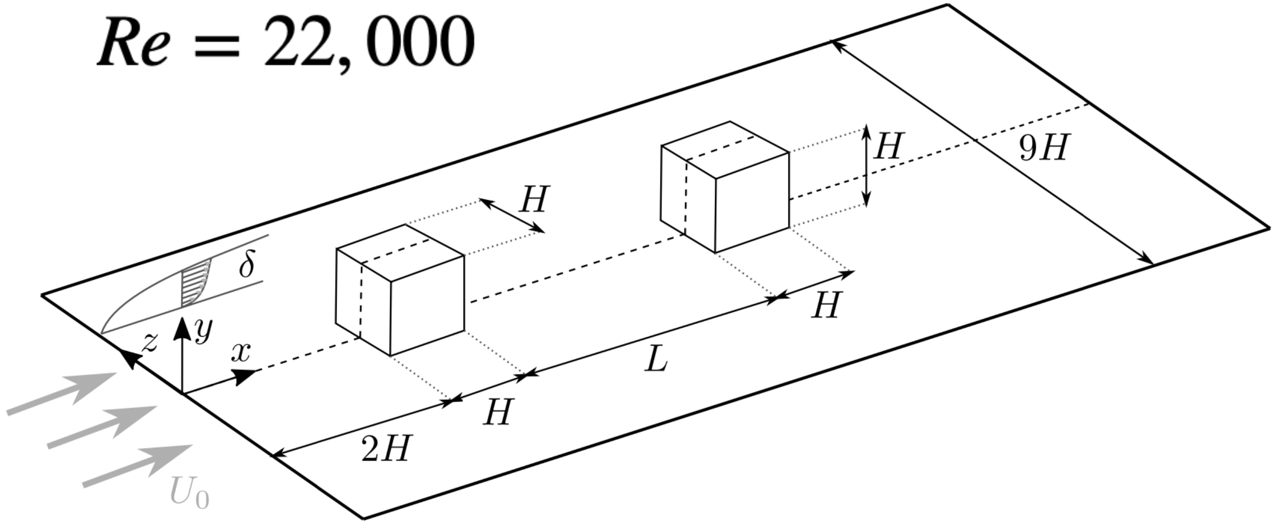
94 invariant scalars derived from U, p, k

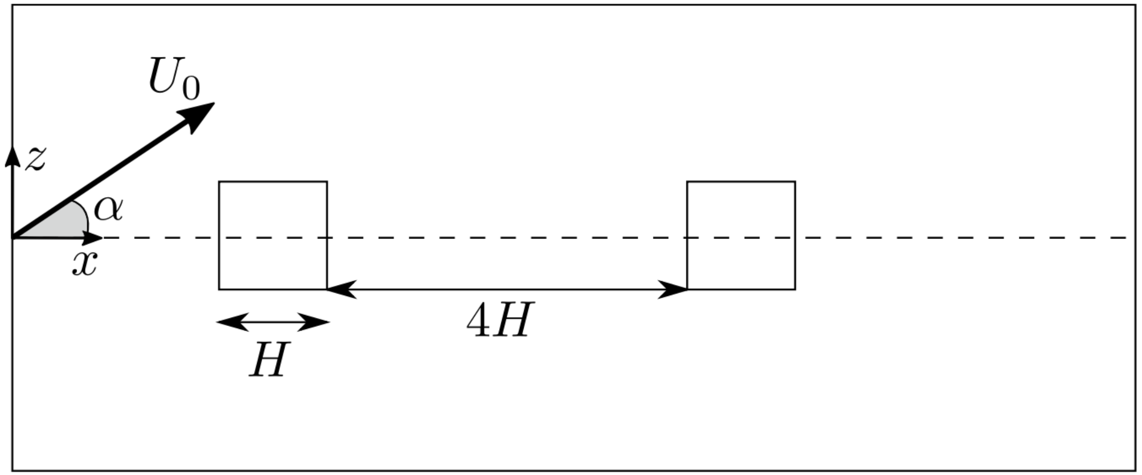
Model Target

Model predicts: $\nu_t^{\dagger}$, $a^{\perp}$

- ***XGBoost*** - gradient boosted decision trees (Chen 2016)
  - Outperforms neural networks for ***tabular data regression*** problems (Shwartz-Ziv 2022)
  - Training performed using ***multi-GPU*** HPC nodes (4xA100), due to large dataset size

3D flow description





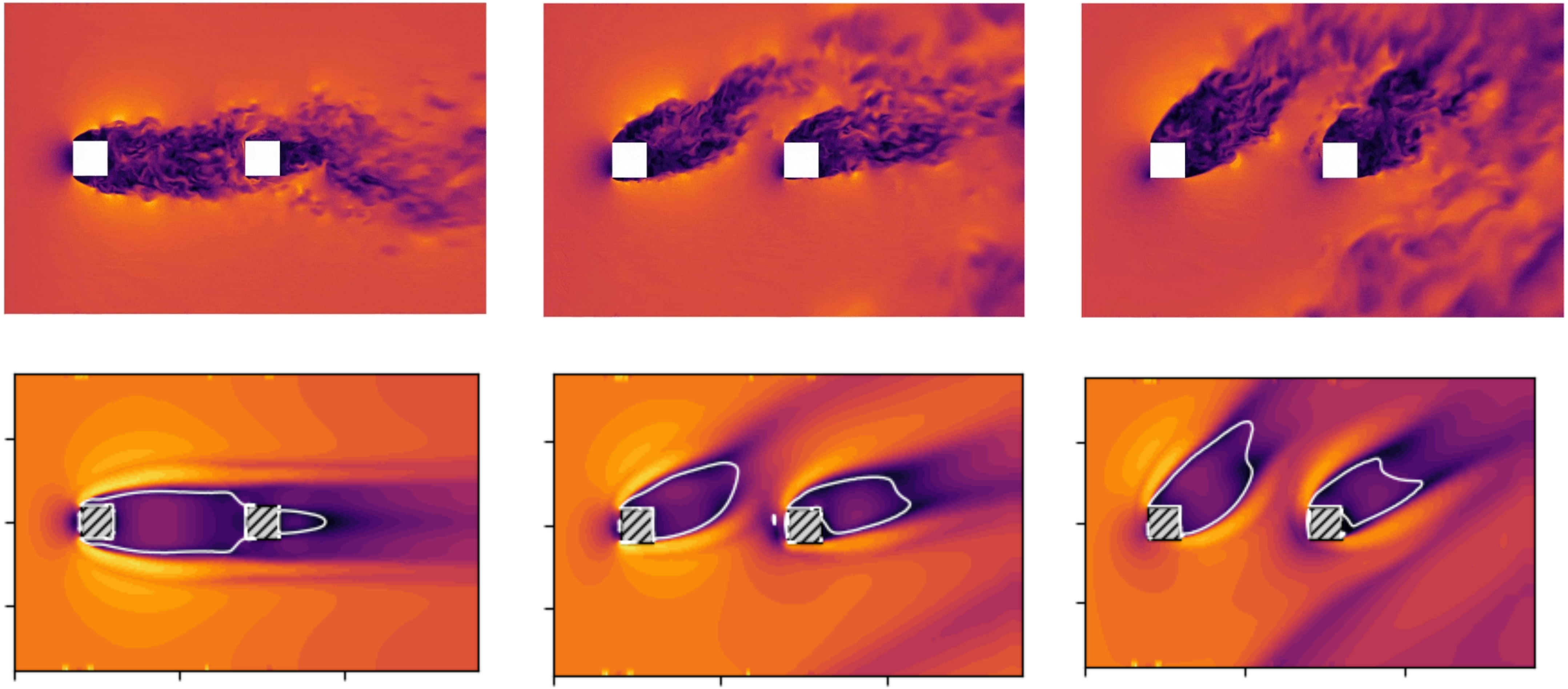
- An array of ***tandem*** wall mounted ***cubes***.
- A range of ***inlet flow angles*** are explored between α=0º and α=45º

Coupling between cubes changes with variations in the parameter space.

Training

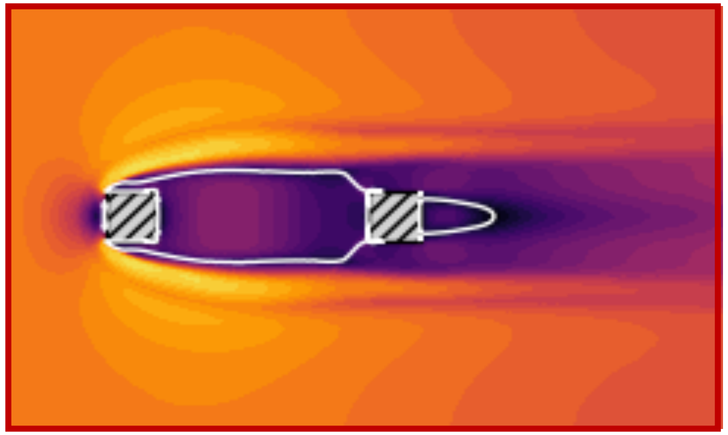
LES dataset

RANS Dataset

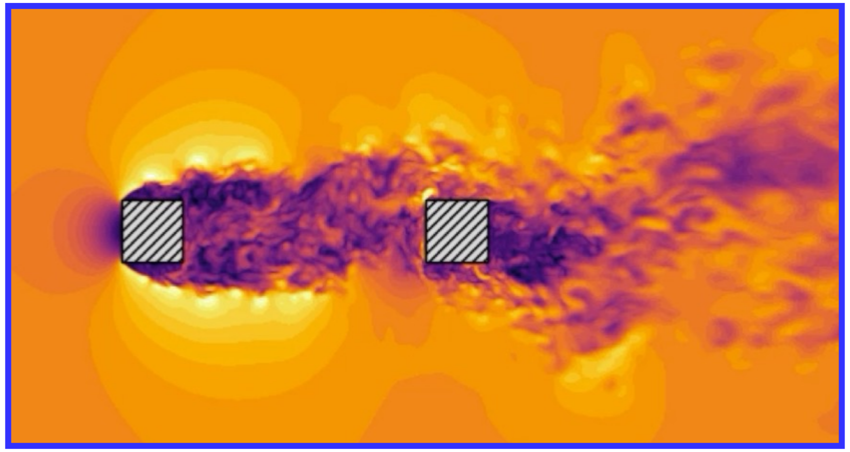


Training

RANS Dataset



LES dataset

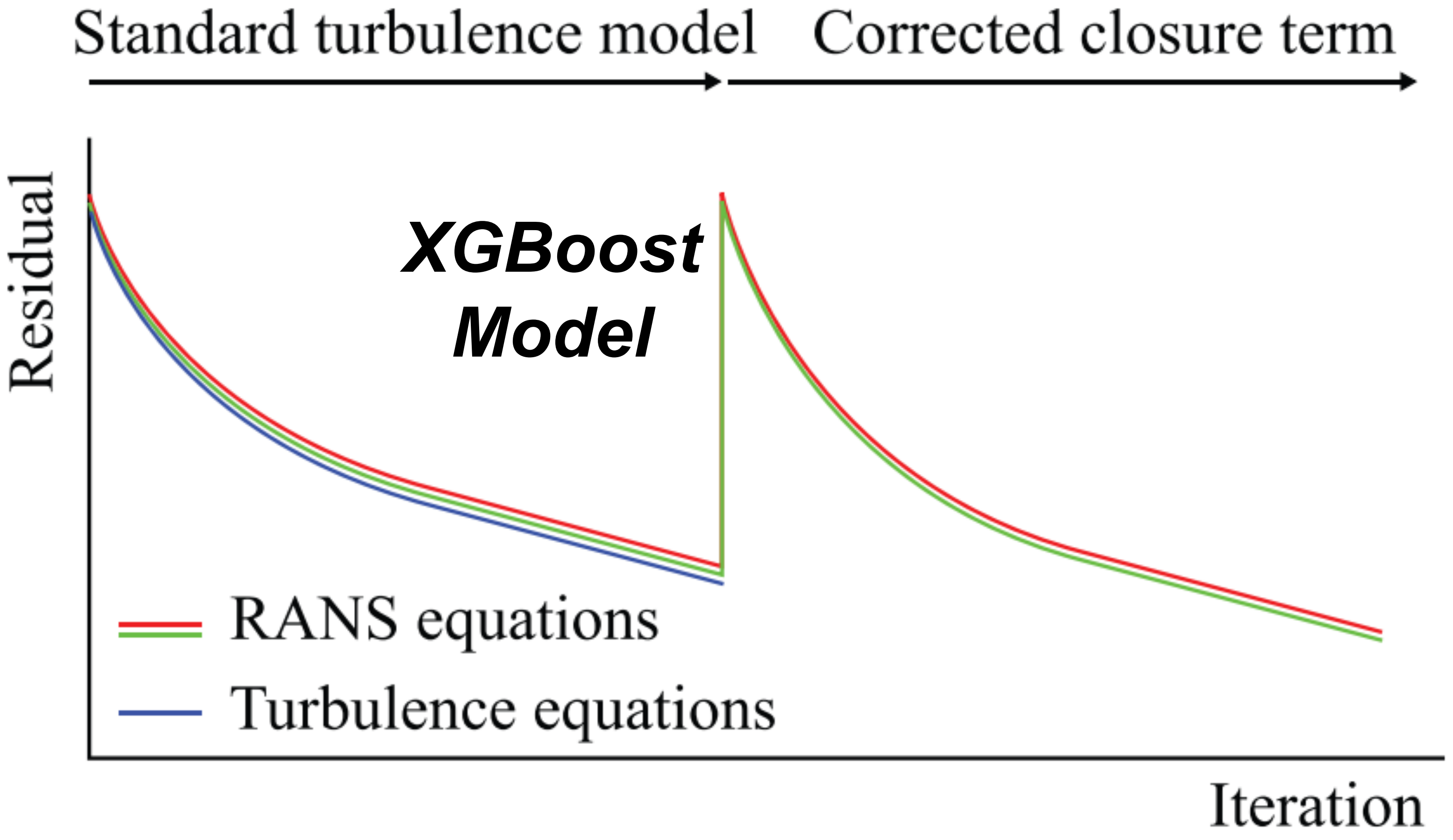


94 invariant scalars derived from U, p, k

XGBoost

Model trained to predict: $\nu_t^{\dagger}$, $a^{\perp}$

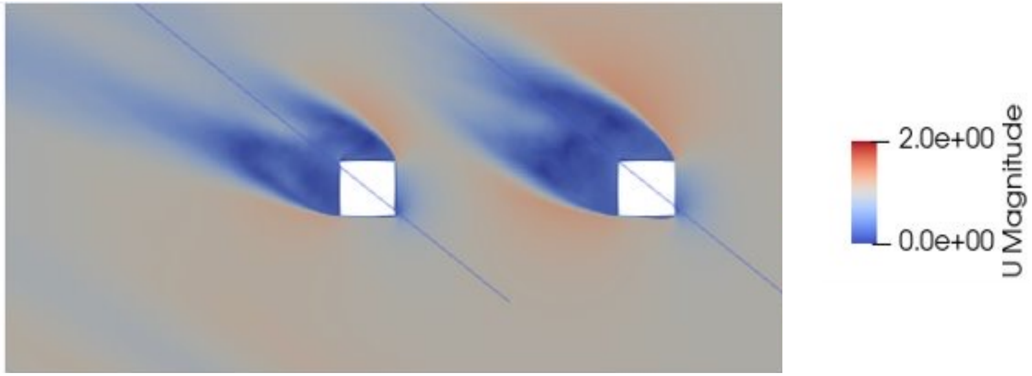
Injection



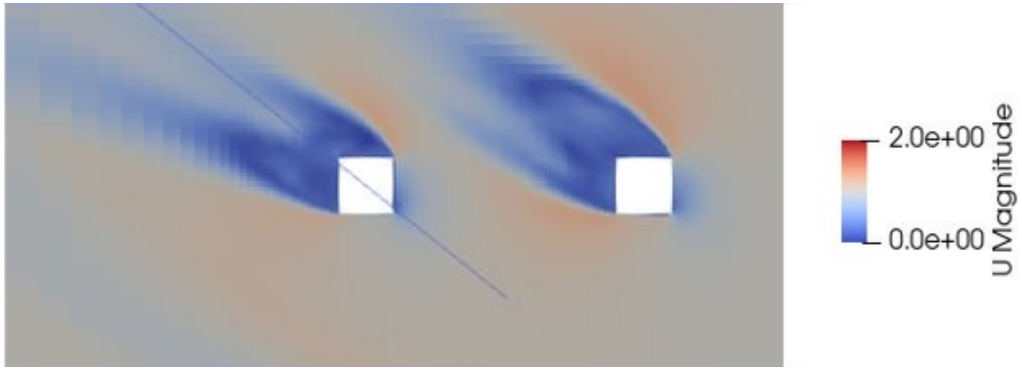
(McConkey et al. 2022) "Qualitative" residual plot showing injection procedure

Results

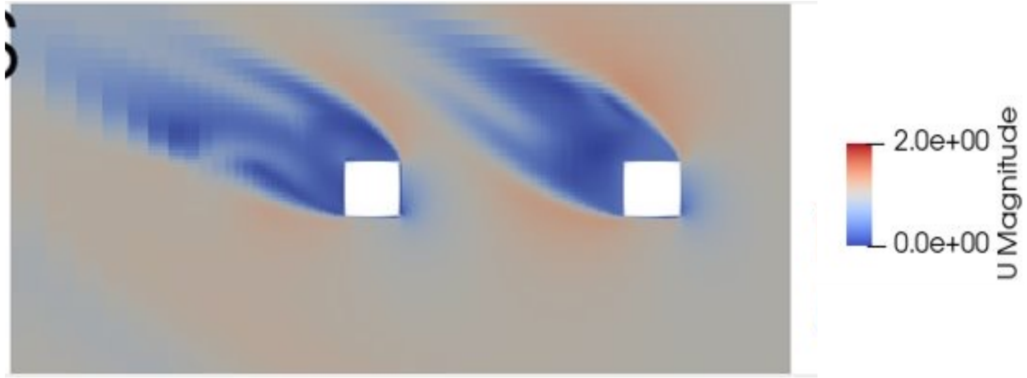
LES



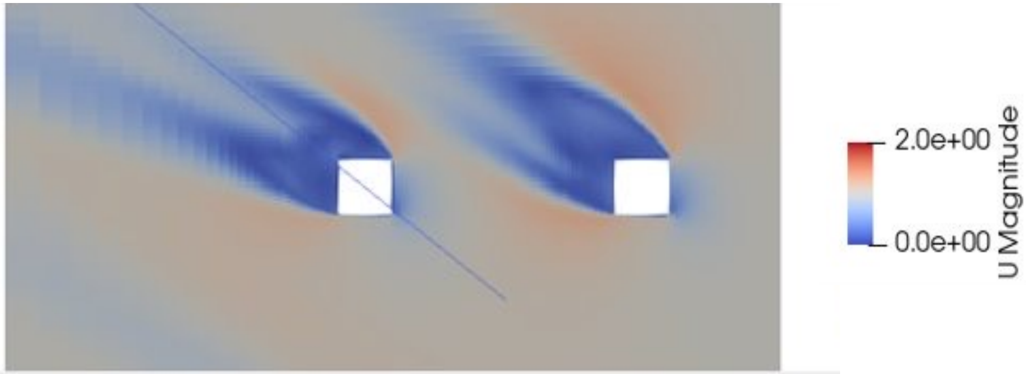
Label Injection



RANS



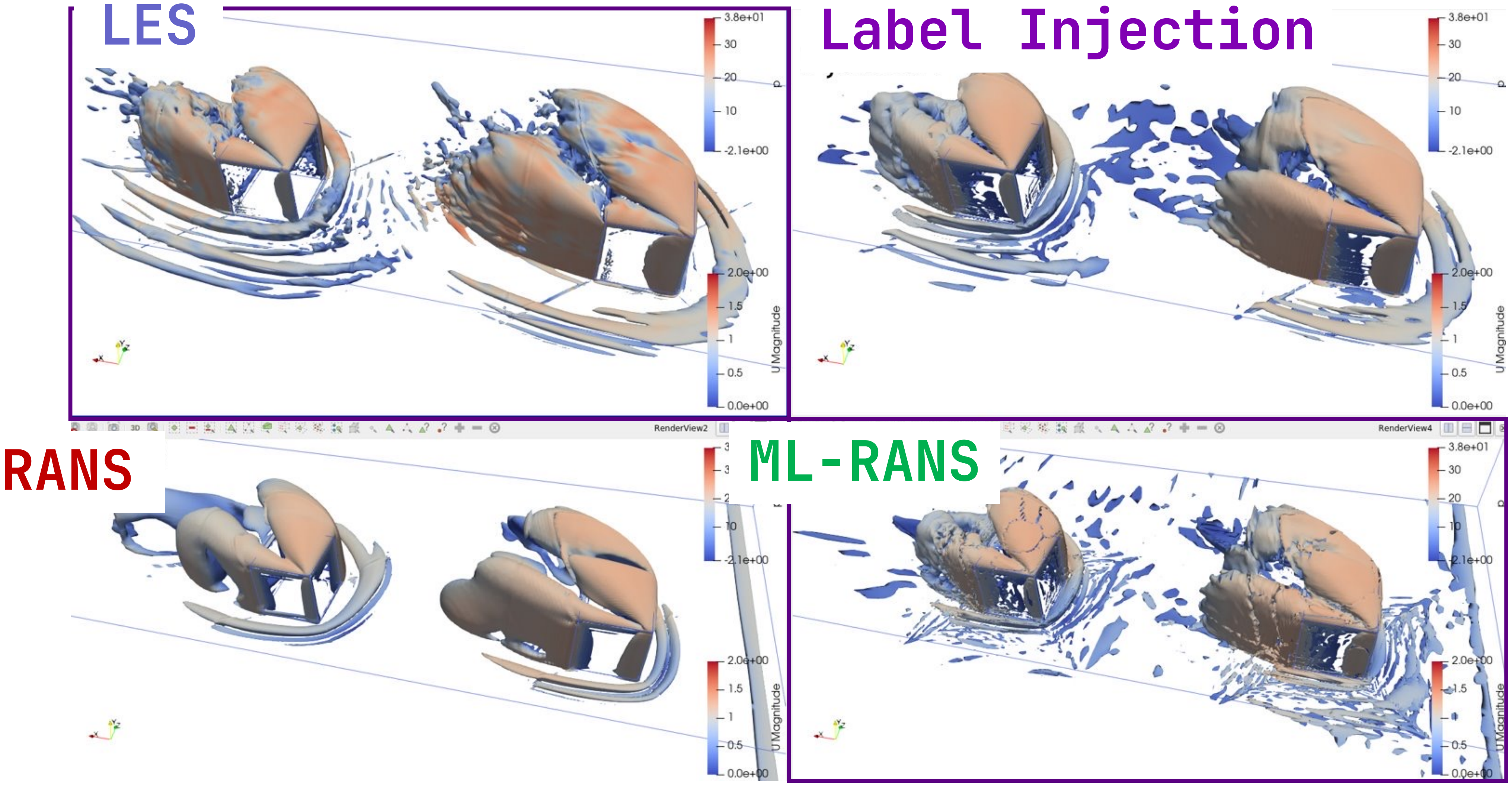
ML-RANS



UNIVERSITY OF WATERLOO | FACULTY OF ENGINEERING

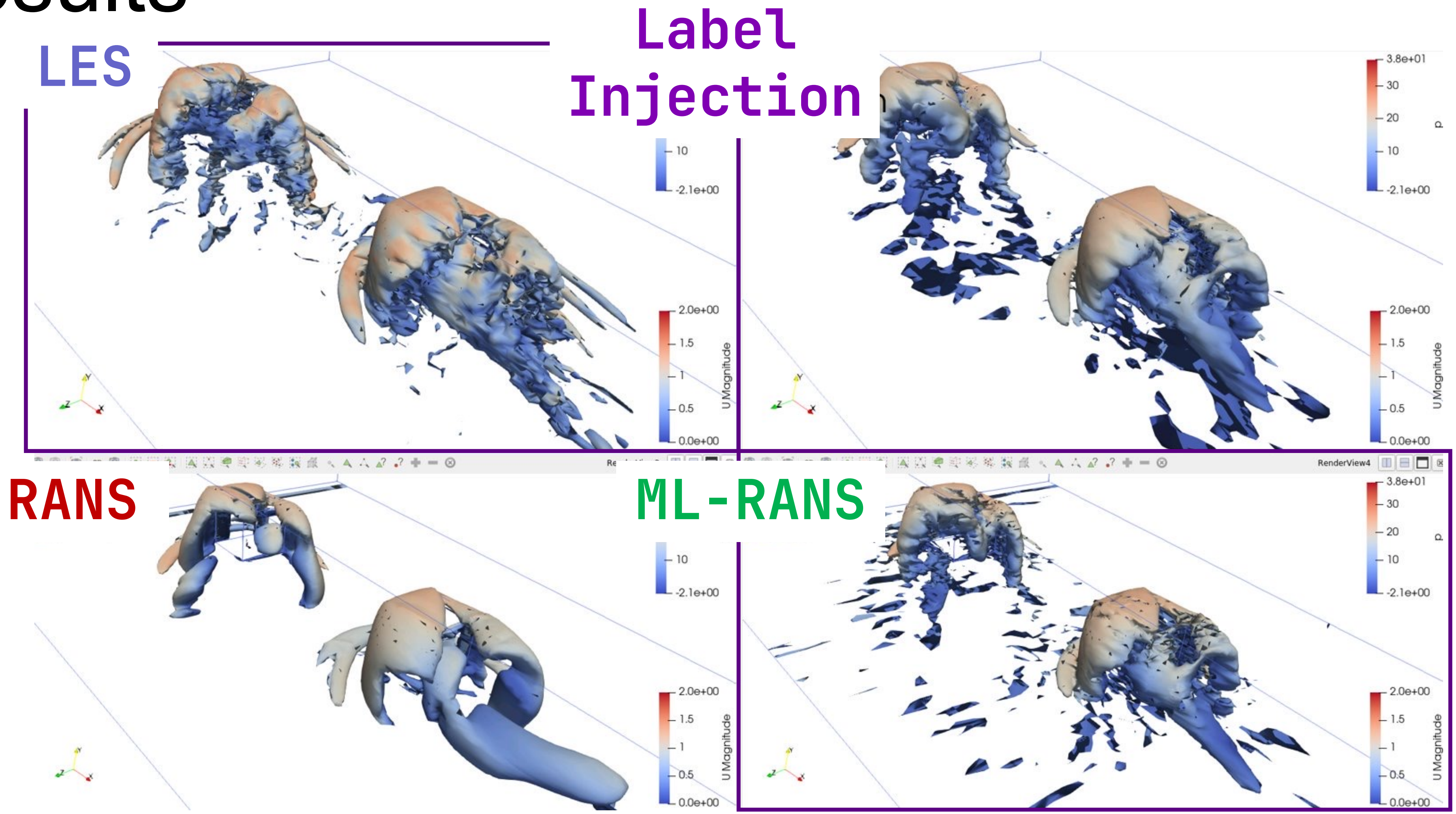
Results

Pressure contours, coloured by velocity magnitude



Results

Pressure contours, coloured by velocity magnitude



Conclusions

- Confirmed applicability and practicality of machine learning techniques for ***industrially relevant***, massively separated ***3D flows***.
- Confirmed ***stability*** of injected XGBoost predictions into RANS equations.
- Developed distributed, ***multi-GPU*** XGBoost training code for training ***large data-driven turbulence models*** using large LES datasets on HPC.

Future work

- Extending framework to ***unsteady*** flows.
- Incorporating XGBoost directly into ***turbulence model***.

References

[1] M. L. A. Kaandorp and R. P. Dwight, "Data-driven modelling of the Reynolds stress tensor using random forests with invariance," *Comput. Fluids*, vol. 202, p. 104497, Apr. 2020, doi: 10.1016/j.compfluid.2020.104497.

[2] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," *Inf. Fusion*, vol. 81, pp. 84–90, 2022, doi: 10.1016/j.inffus.2021.11.011.

[3] B. P. Brener, M. A. Cruz, R. L. Thompson, and R. P. Anjos, "Conditioning and accurate solutions of Reynolds average Navier-Stokes equations with data-driven turbulence closures," *J. Fluid Mech.*, vol. 915, pp. 1–27, 2021, doi: 10.1017/jfm.2021.148.

[4] J. Slotnick, A. Khodadoust, J. Alonso, and D. Darmofal, "CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences," 2014.

[5] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[6] R. McConkey, E. Yee, and F. S. Lien, "Deep structured neural networks for turbulence closure modeling," *Phys. Fluids*, vol. 34, no. 3, p. 035110, Mar. 2022, doi: 10.1063/5.0083074.

[7] F. D. Witherden and A. Jameson, "Future directions of computational fluid dynamics," *23rd AIAA Comput. Fluid Dyn. Conf. 2017*, pp. 1–16, 2017.

[8] J. Ling, A. Kurzawski, and J. Templeton, "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance," *J. Fluid Mech.*, vol. 807, pp. 155–166, 2016, doi: 10.1017/jfm.2016.615.