Adaptive sampling for multi-fidelity neural networks applied to CFD for improved accuracy and interpretability

Harshinee Goordoyal[1,2], Andrew Barnes[1], Andrew Cookson[2], Katharine Fraser[2]

[1] Department of Computer Science, University of Bath, Claverton Down, Bath BA2 7AY, UK

[2] Department of Mechanical Engineering, University of Bath, Claverton Down, Bath BA2 7AY, UK

Abstract.
Background: Computational fluid dynamics simulations range from low-fidelity Reynolds-Averaged Navier-Stokes (RANS) simulations to high-fidelity Direct Numerical Simulations (DNS). While RANS simulations are less computationally expensive, they can be inaccurate. Multi-fidelity machine learning offers the potential to learn the approximation between RANS and DNS and the aim of this study was investigate a suitable sampling strategy.
Methods: A composite multi-fidelity neural network architecture, consisting of a linear and a non-linear branch was used. The method was tested using four canonical flow cases: curved-backward facing step, converging-diverging channel, periodic hill, and parametric bump. Error-based adaptive sampling was introduced to enhance the interpretability and accuracy of the model, by iteratively picking DNS samples that contain the most information. This was compared to random sampling.
Results: Adaptive sampling improves model accuracy compared to random sampling (34% improvement in RMSE with adaptive sampling) and reduces model variability (49% reduction in standard deviation) between repeated training experiments. The selected high-fidelity samples using adaptive sampling were found to be concentrated in regions with recirculation and separation, such as near walls and obstacles, which highlights these regions as the most important for the model to accurately learn the approximation. These results agree with CFD expert knowledge, validating the ability of adaptive sampling to enhance both model performance and interpretability.

1. Introduction

Computational fluid dynamics is a vital tool in many engineering and scientific applications and involves solving the Navier-Stokes equations. Fluid flows exist in three regimes: laminar, transitional and turbulent, determined by the dimensionless quantity Reynolds number. Moving from laminar to turbulent regimes, flow characteristics become more chaotic, with the existence of turbulent eddies and vortices at various spatial and temporal scales. Direct numerical simulations (DNS) solve the full Navier-Stokes equations numerically. For transitional and turbulent cases, this requires mesh elements and time steps small enough to capture the whole turbulent eddy cascade, down to the smallest vortices. To reduce computational cost, Reynolds-Averaged Navier-Stokes Simulations (RANS) are commonly used in practical applications with turbulence. These solve for an averaged flow and apply turbulence models, such as the *k-ω-SST* model [1]. Large Eddy Simulations (LES) lie between the two in terms of computational cost and complexity, resolving for a wide range of temporal and spatial variations of eddies.

In many applications, there is an abundance of easily obtainable low-fidelity RANS data but a lack of high-fidelity DNS/LES data. RANS models are commonly used due to the low computational cost but have been found to be unsuitable for cases where there is curvature, separation or large pressure gradients [2]. However, higher fidelity methods, such as DNS and LES, are prohibitively expensive for multi-objective design, where many simulations are required for a range of design parameters and operating conditions, each of which are computationally expensive due to rotating domains, unsteady flows and turbulence [3].

One alternative to these traditional approaches is machine learning. Various authors have employed different types of multi-fidelity machine learning methods to learn approximations between RANS/LES and DNS, mostly for improved turbulence modelling in the form of either improving RANS solvers by correcting anisotropic stress tensor [4-7] or improving LES by learning sub-grid

scale stresses [8-11]. Meng et al. [12] proposed a composite multi-fidelity neural network architecture to approximate between low-fidelity and high-fidelity data. This method leverages the abundance of low-fidelity data due to lower computational cost and the accuracy of high-fidelity data despite the scarcity. Various authors have applied the composite neural network architecture for multi-fidelity machine learning in CFD, more specifically for aerodynamic shape optimisation [13-15]. While these papers show promising results for applying data-driven multi-fidelity neural networks to learn the approximation between an abundant low-fidelity set and a scarce high-fidelity set, they use a set number of arbitrarily chosen high-fidelity points as training data. Randomness in the chosen samples also means that the method is not reproducible and highly variable based on the random sample set. Error based adaptive sampling, which has been previously used for training machine learning models [16,17], is a potential solution.

The aim of this research was to identify the most important points from a high-fidelity CFD dataset for multi-fidelity machine learning**.** We investigated the correspondence between regions of the input space that are particularly relevant for generating accurate results and the physical flow features.

2. Methods

A multi-fidelity neural network was used to learn the mapping between low-fidelity RANS and high-fidelity DNS/LES data in canonical CFD flow problems. This implied moderately turbulent wall-bounded flows with curvature and separation. Adaptive sampling and random sampling were used to select the high-fidelity points on which the models were trained.

2.1. Dataset

The open-source dataset used to train the model was published McConkey et al. [18] for use in developing machine learning methods for corrective turbulence modelling. The dataset consists of low-fidelity RANS simulation data from four RANS models and the corresponding high-fidelity time-averaged DNS/LES data for each domain and flow condition. The high-fidelity simulations were run on fine grids but the solution in each case was then mapped onto a coarser grid to match the RANS grid. The geometries chosen for this study are in 2D and consisted of the curved backward facing step (CBFS), the periodic hill (PH), the converging-diverging channel (CDC) and the parametric bump (PB) (see figure 3 in the Appendix). All cases considered have flows within the transitional and turbulent regimes, ranging from 2500 to 13 700. The RANS data from two transport-equation RANS turbulence closure model k-ω-SST was used for the investigation.

2.2. Architecture

The multi-fidelity neural network architecture, proposed by Meng et al. [12], was employed in this study to learn the approximation from low-fidelity RANS data to high-fidelity time-averaged DNS/LES data. It is based on the theoretical assumption that a high-fidelity function can be modelled as a function of the corresponding low-fidelity function, more specifically, as the sum of linear $(\mathcal{F}_{\ell})$ and non-linear $(\mathcal{F}_{n\ell})$ components (equation 1).

$$y_H = \mathrm{F}_l(x,y_L) + \mathrm{F}_{nl}(x,y_L) \quad (1)$$

The architecture consists of two branches: the linear branch with no activation and the non-linear branch with non-linear activation (*tanh*). The inputs to the model were the x and y coordinates and the horizontal (U_L) and vertical low-fidelity (V_L) velocity components. The outputs from the linear and non-linear branches were added to give overall outputs of the model U_H (predicted horizontal velocity) and V_H (predicted vertical velocity). The model was trained to minimise the sum of the L2 loss between the model prediction and the high-fidelity data.

2.3. Sampling Strategies

The model was trained on a small number of high-fidelity samples and tested on a large test set of low-fidelity data. The number of points the model was trained on and the location of these points impacts the accuracy of the method. The following two sampling strategies were investigated:

1. Random sampling
   The samples are randomly selected from the domain. The training set is defined at the start of the experiment based on a pre-determined fixed number of maximum samples η_{max} for each geometry.

2. Adaptive sampling
   The adaptive sampling proposed for training the multi-fidelity model is based on an iterative scheme which picks samples with the highest predictive errors after the model has been trained on an interim training set. The pseudo-code for this sampling strategy is shown as algorithm 1.

Algorithm 1: Error-based adaptive sampling
(X: input data, Y : output data, Y_{pred}: model prediction, n: number of samples, m: points with highest errors, η_{max}: maximum number of samples, ϵ_{crit}: error criteria).

Randomly select n initial samples $(x_n, y_n) \in (X, Y)$ without replacement;
Update the dataset (X_{new}, Y_{new});
Split into training and validation sets $([x_{n,train}, y_{n,train}]$ and $[x_{n,val}, y_{n,val}])$;
Initialise multi-fidelity model $\mathcal{M}_{MF}$;
Train $\mathcal{M}_{MF}$ with $[x_{n,train}, y_{n,train}]$;
Predict $Y_{pred} = \mathcal{M}_{\mathcal{MF}}(X_{new})$;
Error $\epsilon = \sum_{n=1}^{i} |Y_{pred} - Y_{new}|$;
while $\epsilon < \epsilon_{crit}$: **do**
  Pick m points with highest $|\epsilon_i|$ from (X_{new}, Y_{new});
  Add m points (x_m, y_m) to (x_n, y_n);
  Update the dataset (X_{new}, Y_{new});
  Split into training and validation sets $([x_{n,train}, y_{n,train}]$ and $[x_{n,val}, y_{n,val}])$;
  Train $\mathcal{M}_{MF}$ with $[x_{n,train}, y_{n,train}]$;
  Predict $Y_{pred} = \mathcal{M}_{\mathcal{MF}}(X_{new})$;
  Error $\epsilon = \sum_{n=1}^{i} |Y_{pred} - Y_{new}|$
end
Save model $\mathcal{M}_{MF}$

The approach used in this study was setting the $\eta_{max} = 4250$ for both random sampling and adaptive sampling. For adaptive sampling, n and m were set to 2000 and 500 respectively, and ϵ_{crit} was set to a very small value. For each case and sampling method, the model training was repeated 20 times to capture the variability in model performance introduced from randomness in the point selection and other sources of nondeterminism which are known to affect model performance [19]. As the experiments were repeated multiple times, the error between the prediction and high-fidelity data was expected to also fluctuate.

2.4. Training and Testing

Initially, the dataset was evenly randomly split into a training and testing set, which were kept consistent across all experiments for uniformity. With both random and adaptive sampling, the models were trained on a training set with up to η_{max} points from this training set. An 80-20 split of the samples was used between training and validation. Section 3 reports the model performance on the unseen test set for each geometry. The number of layers and neurons in each branch were optimised by

grid search using random DNS samples. This was done arbitrarily for the periodic hill case and for simplicity, the best model hyperparameters were used across the other cases. The model was initialised with the Glorot initialization and trained with the Adam optimiser followed by the quasi-newton optimiser L-BFGS.

3. Results and Discussion

From the hyperparameter tuning, the optimum configuration was 2 layers with 32 neurons each in the linear branch and 5 layers with 64 neurons each in the non-linear branch for the periodic hill. Figure 1 and 2 show the flow fields (RANS, Model Prediction, DNS, Error) using adaptive sampling for the PH and CDC respectively. Both cases show low prediction errors throughout the domain. The results from the other cases for both random and adaptive sampling are displayed in the Appendix.

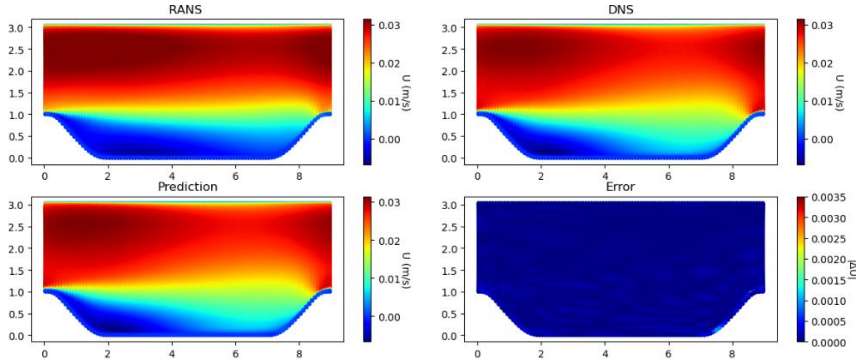


Figure 1: Horizontal velocity flow field over PH (Model prediction (adaptive Sampling), DNS, RANS, Error (absolute))

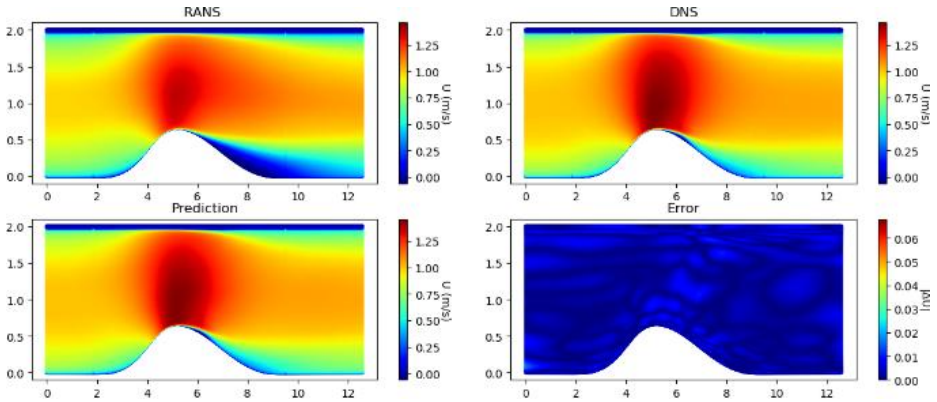


Figure 2: Horizontal velocity flow field over CDC (Model prediction (adaptive Sampling), DNS, RANS, Error (absolute))

Table 1 shows the resulting error distributions from the repeated experiments with random and adaptive sampling, compared to the RANS baseline. The lowest error for each geometry and velocity component is highlighted in bold in table 1. Both adaptive sampling and random sampling produced a reduction in RMSE as compared to the RANS baseline, demonstrating the efficacy of this multi-fidelity method. However, adaptive sampling outperformed random sampling in every case considered. On average, a 34% improvement in RMSE was observed between adaptive and random sampling and an 96% improvement was observed between adaptive sampling and the RANS baseline. In addition, the standard deviation of the error with adaptive sampling was significantly less than with random sampling (49% average improvement in standard deviation), showing less model variability with adaptive sampling.

Adaptive sampling enhances interpretability in scientific computing, particularly for computational fluid dynamics (CFD) and multi-fidelity learning, by strategically selecting training points in high error flow regions. In addition to improving overall model accuracy, it identified the most important regions for refining the mapping from low-fidelity to high-fidelity data. Therefore, adaptive sampling allows the selection of optimal areas for sensor placement for collecting high-fidelity data in an experimental setting. Notably, adaptive sampling concentrated points near zones where accurately capturing flow behaviour is known to be a challenge with low-fidelity CFD (near the walls and around the obstacles), reinforcing confidence in the selection process (see figure 17 in the Appendix). This ensures that these critical regions receive greater attention in the training process. This alignment with fundamental CFD principles and domain knowledge makes the sampled points more reliable and meaningful.

Table 1: Error comparison between RANS, models trained with random sampling, and models trained with adaptive sampling. RMSE U is the root mean squared error between the horizontal velocity from the method (RANS, random or adaptive) and the high-fidelity data, and RMSE V is the same for vertical velocity.

Case	RANS (Baseline)		Random		Adaptive	
	RMSE U	RMSE V	RMSE U	RMSE V	RMSE U	RMSE V
CBFS	8.89E-02	5.78E-03	(4.41 ± 0.61) E-03	(7.22 ± 0.98) E-04	**(2.33 ± 0.46) E-03**	**(3.64 ± 0.50) E-04**
CDC	1.67E-01	3.14E-02	(4.79 ± 0.40) E-03	(1.15 ± 0.11) E-03	**(4.12 ± 0.38) E-03**	**(1.06 ± 0.11) E-03**
PH	2.61E-03	7.97E-04	(1.76 ± 0.32) E-04	(8.58 ± 1.29) E-05	**(9.80 ± 1.03) E-05**	**(6.29 ± 0.52) E-05**
PB	2.23E+00	5.06E-01	(1.04 ± 0.20) E-01	(3.03 ± 0.82) E-02	**(5.62 ± 0.20) E-02**	**(1.97 ± 0.06) E-02**

4. Conclusion and Future Work

In this study, multi-fidelity neural networks were applied to canonical fluid flow problems to learn the approximation between low-fidelity RANS data and high-fidelity DNS or LES data. The work focuses on two sampling methods: random and adaptive sampling, and how they affect accuracy and interpretability of the machine learning model. The results show that training the model with both random and adaptive sampling (with $\eta_{max} = 4250$) from the high-fidelity dataset produces more accurate flow predictions than the low-fidelity RANS baseline. The adaptive sampling scheme prioritises the selection of points showing high errors during the training process, producing an improvement compared to random sampling (34%). Adaptive sampling also enables a reduction in model variability, as evidenced by a lower standard deviation (49% reduction) in the reported error on the test set. This demonstrates that adaptive sampling is an important tool that can be used with data-driven multi-fidelity neural networks. Furthermore, the use of adaptive sampling sheds light on which regions in the domain provide the most information and therefore are the most important for learning the approximation from low to high-fidelity. Regions around the obstacles or near the walls are often chosen by adaptive sampling, highlighting them as potential zones for positioning sensors for high-fidelity data collection. Therefore, adaptive sampling in this context improves both accuracy and interpretability.

Acknowledgments

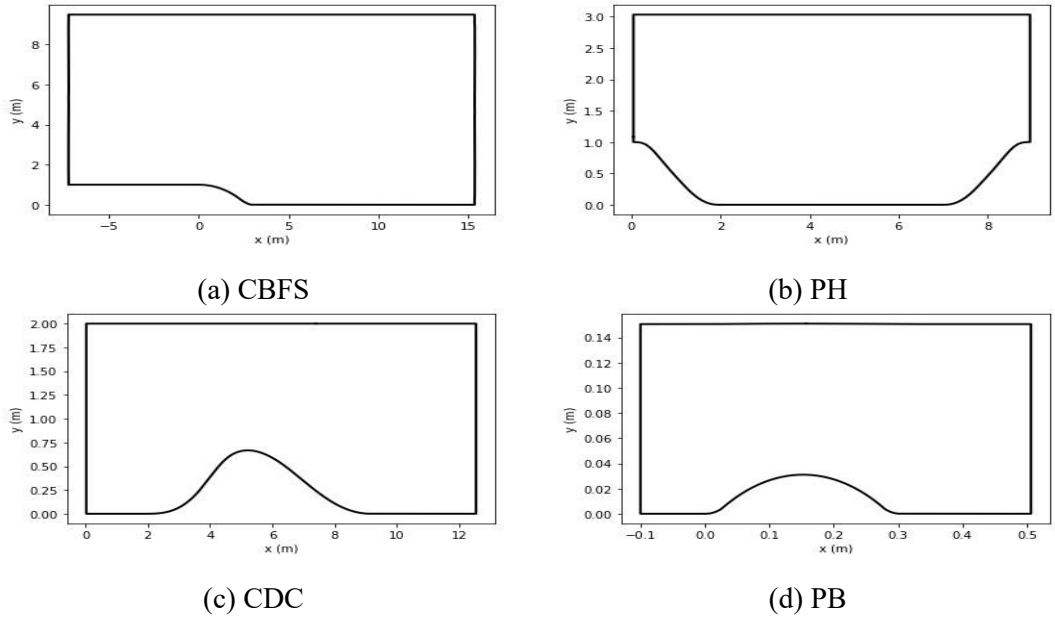
This research made use of Hex, the GPU Cloud in the Department of Computer Science at the University of Bath. This work was supported by the UKRI Centre for Doctoral Training in Accountable, Responsible and Transparent AI (ART-AI) [EP/S023437/1] and the University of Bath.

References

[1] Florian R Menter. Improved two-equation k-omega turbulence models for aerodynamic flows. Technical report, 1992.
[2] TJ Craft, BE Launder, and K Suga. Development and application of a cubic eddy-viscosity model of turbulence. *International Journal of Heat and Fluid Flow*, 17(2):108–115, 1996.
[3] Peng Wu. Recent advances in the application of computational fluid dynamics in the development of rotary blood pumps. *Medicine in Novel Technology and Devices*, 16:100177, 2022.
[4] Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155– 166, 2016.
[5] Mikael LA Kaandorp and Richard P Dwight. Data-driven modelling of the reynolds stress tensor using random forests with invariance. *Computers & Fluids*, 202:104497, 2020.
[6] Chao Jiang, Ricardo Vinuesa, Ruilin Chen, Junyi Mi, Shujin Laima, and Hui Li. An interpretable framework of data-driven turbulence modeling using deep neural networks. *Physics of Fluids*, 33 (5), 2021.
[7] Jinxing Li, Yunzhu Li, Tianyuan Liu, Di Zhang, and Yonghui Xie. Multi-fidelity graph neural network for flow field data fusion of turbomachinery. *Energy*, 285:129405, 2023.
[8] Masataka Gamahara and Yuji Hattori. Searching for turbulence models by artificial neural network. *Physical Review Fluids*, 2(5):054604, 2017.
[9] Haochen Li, Yaomin Zhao, Jianchun Wang, and Richard D Sandberg. Data-driven model development for large-eddy simulation of turbulence using gene-expression programing. *Physics of Fluids*, 33(12), 2021.
[10] Andrea Beck, David Flad, and Claus-Dieter Munz. Deep neural networks for data-driven les closure models. *Journal of Computational Physics*, 398:108910, 2019.
[11] Justin Sirignano, Jonathan F MacArt, and Jonathan B Freund. Dpm: A deep learning pde augmentation method with application to large-eddy simulation. *Journal of Computational Physics*, 423: 109811, 2020.
[12] Xuhui Meng and George Em Karniadakis. A composite neural network that learns from multifidelity data: Application to function approximation and inverse pde problems. *Journal of Computational Physics*, 401:109020, 2020.
[13] Lei He, Weiqi Qian, Tun Zhao, and Qing Wang. Multi-fidelity aerodynamic data fusion with a deep neural network modeling method. *Entropy*, 22(9):1022, 2020.
[14] Xinshuai Zhang, Fangfang Xie, Tingwei Ji, Zaoxu Zhu, and Yao Zheng. Multi-fidelity deep neural network surrogate model for aerodynamic shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 373:113485, 2021.
[15] Peng Liao, Wei Song, Peng Du, and Hang Zhao. Multi-fidelity convolutional neural network surrogate model for aerodynamic optimization based on transfer learning. *Physics of Fluids*, 33(12), 2021.
[16] Maria Boettcher, Alexander Fuchs, Ferenc Leichsenring, Wolfgang Graf, and Michael Kaliske. Elsa: An efficient, adaptive ensemble learning-based sampling approach. *Advances in Engineering Software*, 154:102974, 2021.
[17] Yi-Ping Chen, Liwei Wang, Yigitcan Comlek, and Wei Chen. A latent variable approach for nonhierarchical multi-fidelity adaptive sampling. *Computer Methods in Applied Mechanics and Engineering*, 421:116773, 2024.
[18] Ryley McConkey, Eugene Yee, and Fue-Sang Lien. A curated dataset for data-driven turbulence modelling. *Scientific data*, 8(1):255, 2021.
[19] Cecilia Summers and Michael J Dinneen. Nondeterminism and instability in neural network optimization. In *International Conference on Machine Learning*, pp. 9913–9922. PMLR, 2021.

Appendix

A.1 Geometries



(a) CBFS (b) PH

(c) CDC (d) PB

Figure 3: The geometries chosen from McConkey et al.'s dataset [18]

A.2 Flow fields (Random Sampling)

A.1.1 Curved Backward Facing Step

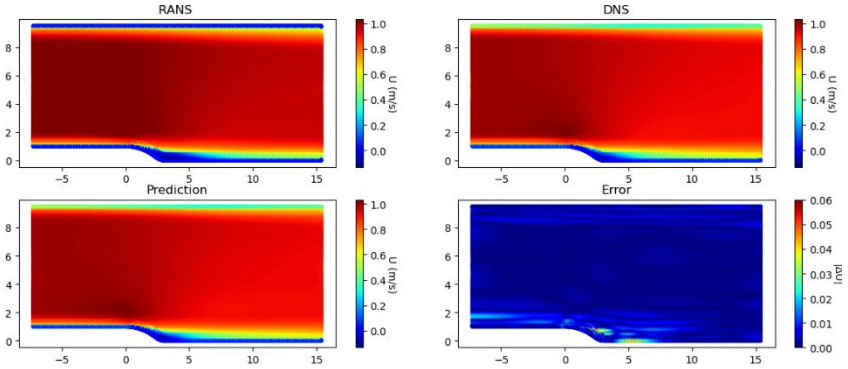


Figure 4: Horizontal velocity flow field over CBFS (Model prediction (Random Sampling), DNS, RANS, Error (absolute))

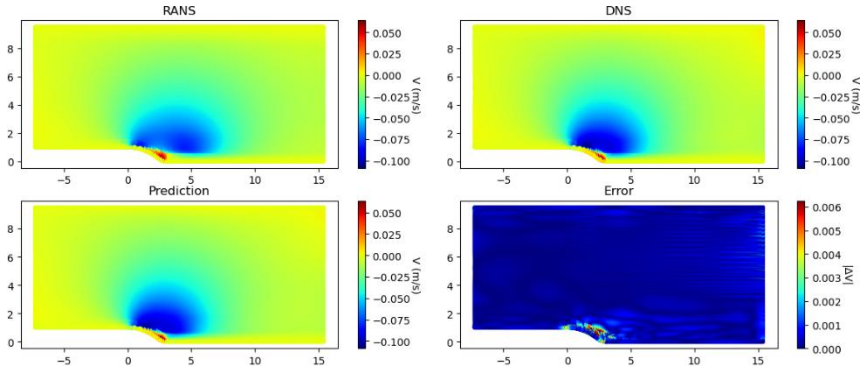


Figure 5: Vertical velocity flow field over CBFS (Model prediction (Random Sampling), DNS, RANS, Error (absolute))

A.1.2 Converging Diverging Channel

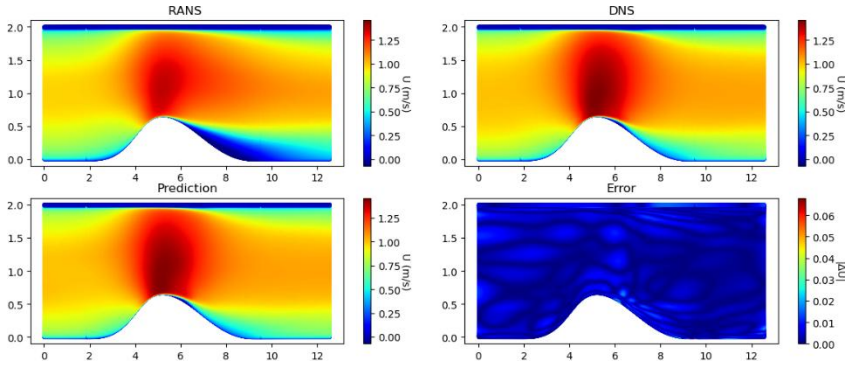


Figure 6: Horizontal velocity flow field over CDC (Model prediction (Random Sampling), DNS, RANS, Error (absolute))

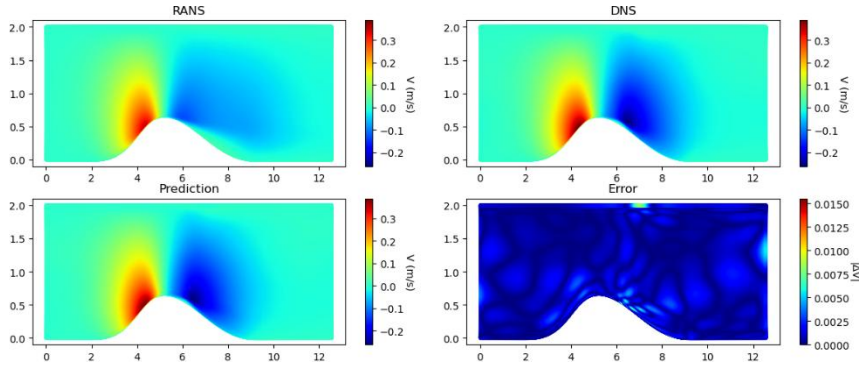


Figure 7: Vertical velocity flow field over CDC (Model prediction (Random Sampling), DNS, RANS, Error (absolute))

A.1.3 Periodic Hill

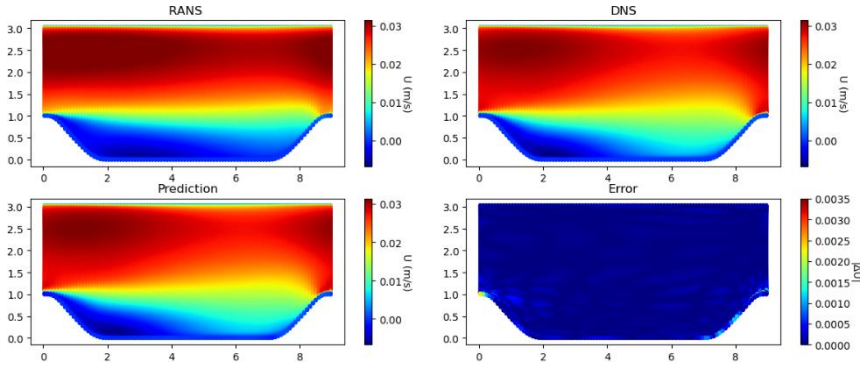


Figure 8: Horizontal velocity flow field over PH (Model prediction (Random Sampling), DNS, RANS, Error (absolute))

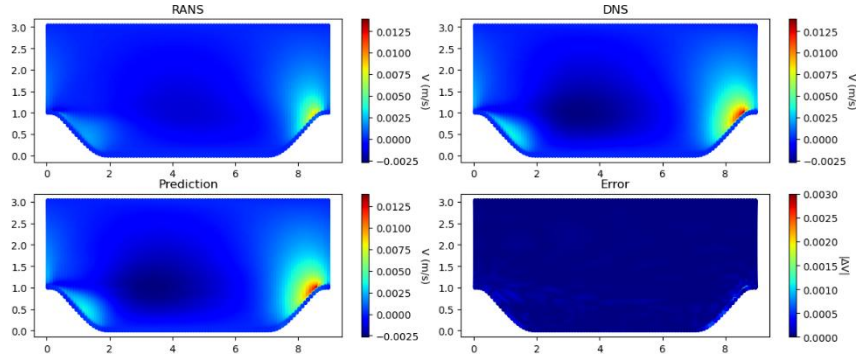


Figure 9: Vertical velocity flow field over PH (Model prediction (Random Sampling), DNS, RANS, Error (absolute))

A.1.4 Parametric Bump

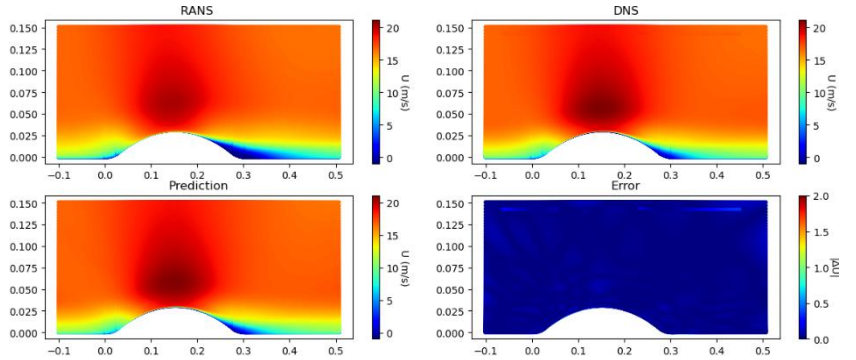


Figure 10: Horizontal velocity flow field over PB (Model prediction (Random Sampling), DNS, RANS, Error (absolute))

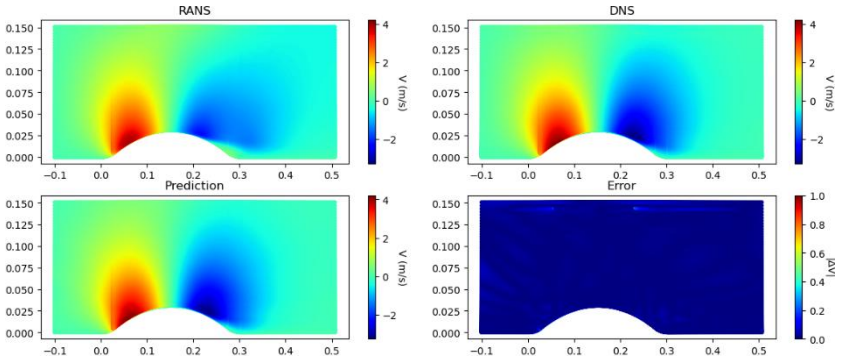


Figure 11: Vertical velocity flow field over PB (Model prediction (Random Sampling), DNS, RANS, Error (absolute))

A.2 Flow fields (Adaptive Sampling)

A.2.1 Curved Backward Facing Step

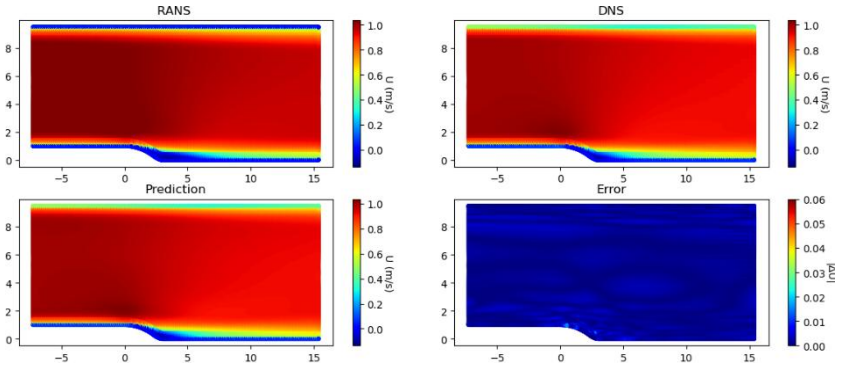


Figure 12: Horizontal velocity flow field over CBFS (Model prediction (adaptive Sampling), DNS, RANS, Error (absolute))

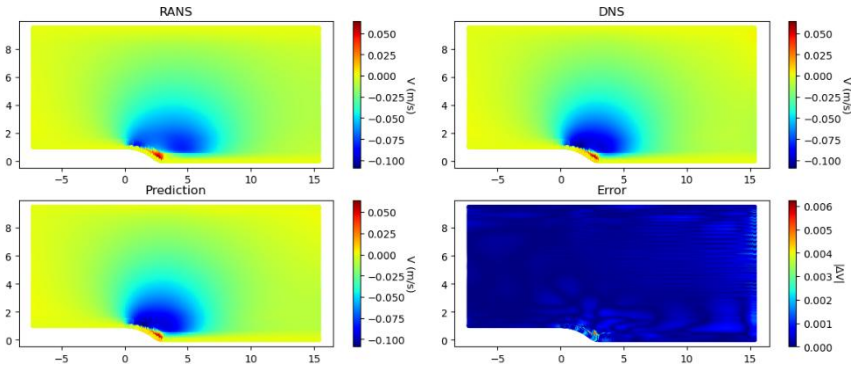


Figure 13: Vertical velocity flow field over CBFS (Model prediction (adaptive Sampling), DNS, RANS, Error (absolute))

A.2.2 Converging Diverging Channel

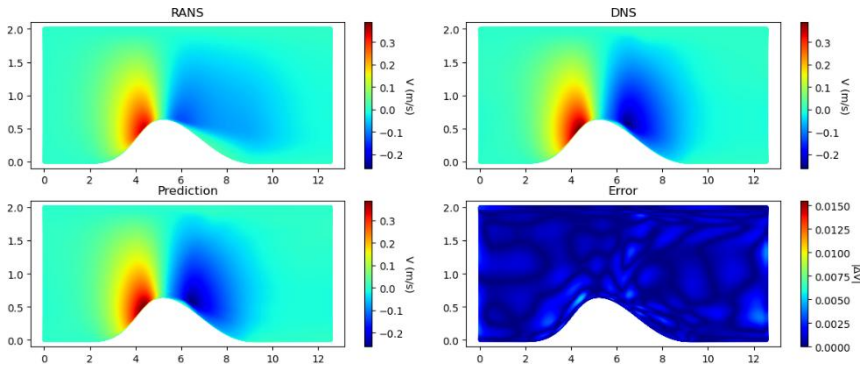


Figure 14: Vertical velocity flow field over CDC (Model prediction (adaptive Sampling), DNS, RANS, Error (absolute))

A.2.3 Periodic Hill

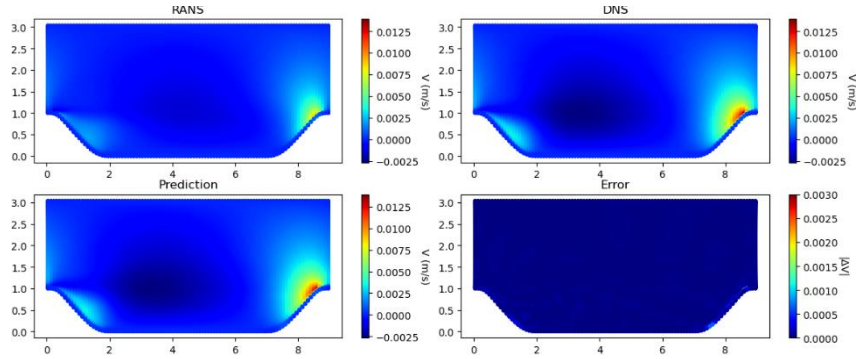


Figure 15: Vertical velocity flow field over PH (Model prediction (adaptive Sampling), DNS, RANS, Error (absolute))

A.2.4 Parametric Bump

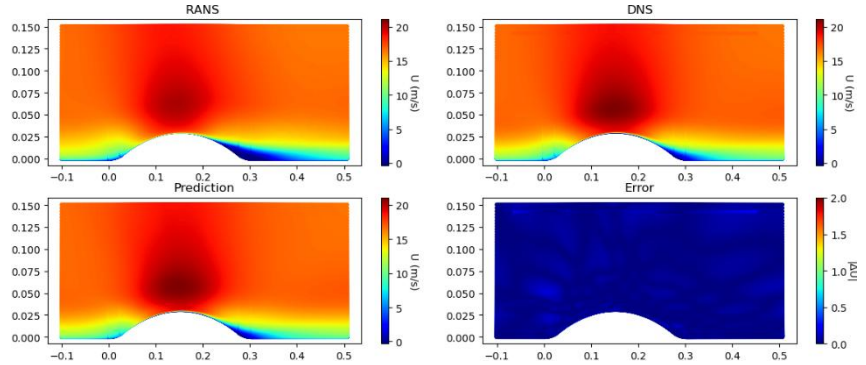


Figure 16: Horizontal velocity flow field over PB (Model prediction (adaptive Sampling), DNS, RANS, Error (absolute))

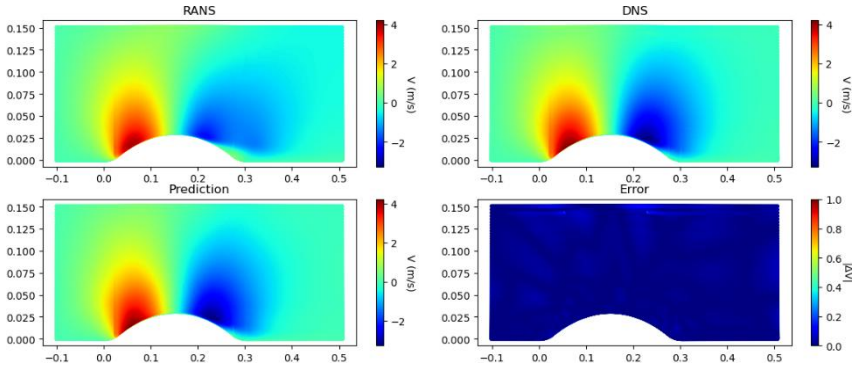
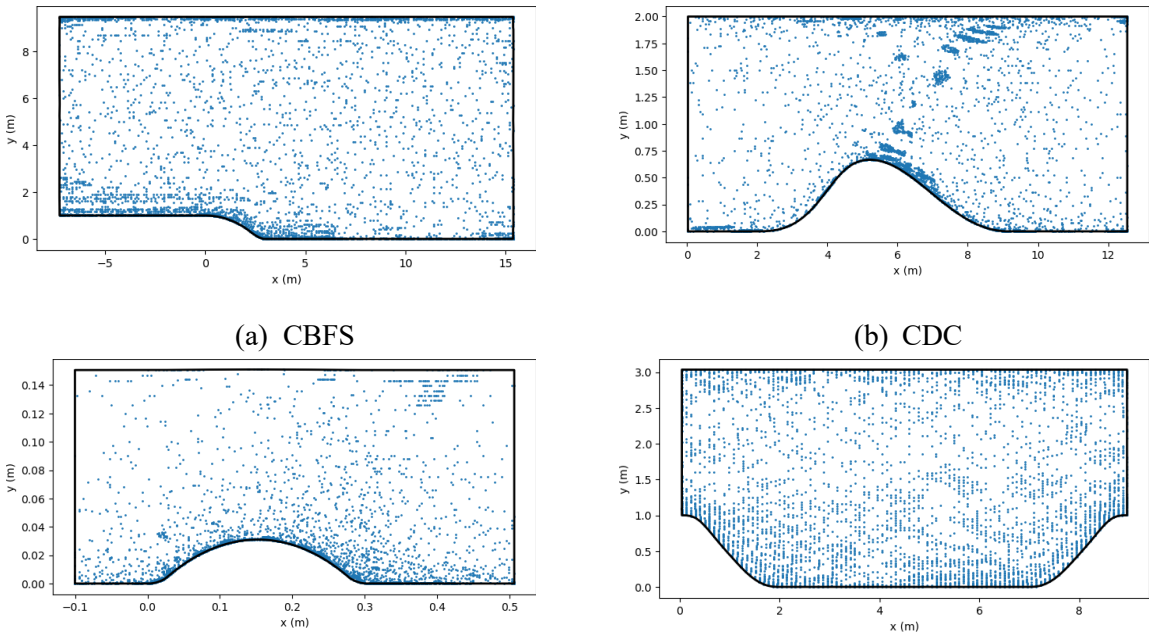


Figure 17: Vertical velocity flow field over PB (Model prediction (adaptive Sampling), DNS, RANS, Error (absolute))

A.3 Location of samples (adaptive)



(a) CBFS (b) CDC

(c) PB (d) PH

Figure 17: Location of samples picked using adaptive sampling across the four domains