
A data-driven wall model for the prediction of turbulent flow separation over periodic hills

Margaux Boxho^{*1,2,3}, Koen Hillewaert^{2,1}, Grégoire Winckelmans³, Michel Rasquin¹,
Thomas Toulorge¹ and Sophie Mouriaux⁴

¹Cenaero, Rue des Frères Wright, 29, 6041 Gosselies, Belgium.

²Aerospace and Mechanics Department (A&M), Université de Liège (ULiège), 4000 Liège, Belgium.

³Institute of Mechanics, Materials and Civil Engineering (IMMC), Ecole Polytechnique de Louvain, 1348 Louvain-la-Neuve, Belgium.

⁴Safran Tech, Rue des jeunes bois, Châteaufort, 7877 Magny-les-Hameaux, France.

Abstract

Aeronautic flows are characterized by turbulence, which consists of chaotic perturbations around a time-averaged flow field. Turbulence appears in a large variety of length scales, ranging from unsteady flow features of the size of the aircraft, wing, blade down to tiny whirls, which are many orders of magnitude smaller. Turbulence has a profound impact on aerodynamic performance, but unfortunately, explicit computation of all turbulent features remains intractable for the design and analysis of real-life geometries [1]. The smallest structures, requiring the largest computational effort, are found in the so-called boundary layer near the wall. This cost can be avoided by modeling their time-averaged impact on the forces exchanged between fluid and the wall. This saving, in turn, allows for the direct computation of the largest turbulent flow features away from the wall, which govern important large-scale effects. The present work proposes the use of Deep Neural Networks (DNN) to link the wall shear stress components to volume data extracted at multiple wall-normal distances h_{wm} and wall-parallel locations. The model focuses on separation since this phenomenon is currently not well-represented, whereas it has a huge impact on aerodynamic performance and operating range. The model is trained using a high-fidelity database of the well-known two-dimensional periodic hill flow. The conditions of this separated flow are such that it is still affordable to compute all turbulent flow features directly, using Tier-1 modern supercomputers.

1 Introduction

Aeronautic flows are characterized by turbulence which consists of an irregular chaotic motion superimposed upon the mean flow field. The turbulent flow features – also called *whirls* or *eddies* – occur over a large range of length scales. The largest eddies are formed by the flow around the geometry. Big eddies are unstable and break down into smaller eddies, transferring almost all of their kinetic energy. The newly formed smaller eddies then undergo the same process. This decomposition occurs in a recursive manner such that kinetic energy is continuously transferred from larger to ever-smaller eddies until finally, the viscous effects dominate where the kinetic energy is converted into heat. The corresponding small scales are often many orders of magnitude smaller than the largest; this ratio grows with the non-dimensional Reynolds number $Re = uD/\nu$ with u , D , and ν , a representative velocity, length scale, and viscosity, respectively. This process is commonly referred to as the *energy cascade*, introduced by Richardson and further refined by Kolmogorov. Turbulence

*Corresponding author margaux.boxho@cenaero.be

occurs in many natural phenomena (smoke, waterfalls, winds, ...) and many important engineering applications (wind loads on buildings, wind turbines, vehicles, aeronautics, ...). The comprehension and characterization of turbulence are of great interest in the aeronautical sector for the design of aircraft, wings, or jet engine components. Due to computational cost considerations, industrial simulation tools rely on the *Reynolds-Averaged Navier-Stokes (RANS)* approach, which replaces the explicit representation of turbulence by modeling its averaged impact on the mean flow. RANS models frequently fail, due to their inherent modeling assumptions, to predict off-design operating points where undesired flow patterns are significantly exacerbated and flow separation may occur. To enable off-design performance prediction, the main challenge is to provide a computational approach with fewer modeling assumptions while maintaining a reasonable computational cost. The direct computation of all scales of turbulence up to the smallest is called *Direct Numerical Simulations (DNS)*. DNS removes all reliance on modeling but remains far too computationally expensive except for very simple geometries and flow conditions. The *Large Eddy Simulation (LES)* approach replaces the explicit representation of the smallest scales by modeling their impact on the computed larger scales. However, the saving that this brings may still not be enough to tackle real-life problems. One can go one step further, considering that the smallest turbulent flow structures are found in a small region – the boundary layer – near the wall. For a commercial aircraft wing, these flow features are 10^8 to 10^9 smaller than the wing section itself. Since *Wall-resolved LES (wrLES)* computes these microscopic whirls in the boundary layer, it still requires prohibitively extensive computational resources [1]. *Wall modeled LES (wmLES)* avoids this cost by modeling the impact of the turbulence in the boundary layer on the forces, and in particular, the wall shear stress exchanged between the flow and the wall. This approach allows economizing computational resources without compromising resolution in the core flow. Literature is abundant concerning wall model proposals (see Larsson et al. [2]): algebraic wall models, two-layer wall models, equilibrium and non-equilibrium wall models, wall models based on partial differential equations, ... to name a few. Most wall models have very restrictive model assumptions, which are not often met in practice. In particular, once the flow separates or transitions, these hypotheses no longer hold.

The purpose of this work is to address the prediction of wall shear stress for separated flows using *deep neural networks (DNN)*. Given the difficulty of representing the complex behavior of the boundary layer in a large range of conditions, DNN offer unique opportunities due to their universal approximation properties Hornik [3]. In literature, the application of DNN to wall modeling has already been partially addressed. Yang et al. [4] attempt to develop a data-driven wall model for the extrapolation at high Reynolds numbers. To improve the extrapolation capabilities, *a priori* physical knowledge is incorporated into the model inputs. Zhou et al. [5] targeted separating turbulent flows by considering the periodic hill problem. Although their *a priori* validation on different geometries and bulk Reynolds gives encouraging results, the *a posteriori* validation on the nominal geometry at $Re = 10595$ fails to predict the mean separation lines leading to an incorrect prediction of the mean velocity profiles. Lozano-Durán and Bae [6] consider multiple flow configurations including corner vortex flows and separation. They assumed that a flow is a non-linear combination of known *building-block flows*. Hence, their wall-flux-based wall model for LES using a self-critical machine-learning approach was trained over DNS data, including flow over a flat plate, in a channel, in a turbulent duct, or separated flow at various Reynolds numbers. The obtained model can not predict the separation in the NASA Juncture Flow, probably due to a lack of data. In our approach, we trained more advanced architecture (such as a Gaussian Mixture Neural network) on an enlarged domain of dependency to better capture the complex relationship between instantaneous flow fields, geometrical parameters, and wall shear stress.

2 The two-dimensional periodic hill

The data-driven model is trained using high resolution wrLES computations of the well-known doubly periodic hill case (see Figure 3), which is described by Fröhlich et al. [7] and part of the well-known ERCOFTAC Knowledge Base Wiki test cases [8]. This bi-periodic flow evolves between two walls featuring a streamwise constriction, inducing turbulent separation and reattachment. The considered bulk Reynolds number is $Re_b = 10595$, whereby $Re_b = \frac{u_b h}{\nu}$, with u_b the bulk velocity measured at the hill top, h the hill height and ν the kinematic viscosity. A constant pressure gradient drives the flow whose magnitude is controlled to match the bulk Reynolds number following Benocci and Pinelli [9]. wrLES results are obtained using a high-order compressible DG flow solver. The flow simulation is performed at a low bulk Mach number M_b of 0.1, and statistics are accumulated over

38 flow-through times ($t_c = u_b/L_x$, where L_x is the streamwise domain length), after evacuating the numerical transient over $45 t_c$. The averaged results, listed in appendix A for the sake of brevity, are compared to those obtained by Temmerman et al. [10] and Gloerfelt and Cinnella [11]. The flow presents a precursory separation at the top of the hill, only present for compressible flows, a second main separation from $x/h \simeq 0.19$ to $x/h \simeq 4.21$ and a third at the leeward base of the hill (see Figure 4). Figure 4 shows the envelope of the time-varying friction coefficient. This graph, combined with the intermittency ($\gamma = t_{(\tau_w < 0)}/t_{tot}$), highlights the variability of the separation location and conditions in time. Around the mean separation line ($x/h = 0.19$), the flow is 40% to 60% of the time detached. According to this observation, the data-driven wall model is required to be correct instantaneously in addition to being correct on average.

3 Data-driven wall-model

Formulation. The prediction of the flow-aligned and transverse wall shear stress ($\tau_{w,\xi}, \tau_{w,z}$) can be stated as a regression problem. Under the assumption of Gaussian distribution, $\mathcal{N}(\mu, \sigma)$ of mean μ and variance σ , where the approximation function f is parametrized using a *neural network (NN)* that contains no activation function on its last layer, and applying the maximum likelihood estimation, we recover the mean squared error loss, where θ represents the neural network parameters (weight and bias), τ_w the vector of output labels, \mathbf{x}_i the input labels and \mathbf{d} the set of i.i.d. training data sets:

$$p(\tau_w|\mathbf{x}) = \mathcal{N}(\tau_w; \mu = f(\mathbf{x}; \theta), \sigma^2 = 1) \Rightarrow \arg \min_{\theta} \sum_{\mathbf{x}_i, \tau_w^i \in \mathbf{d}} (\tau_w^i - f(\mathbf{x}_i; \theta))^2. \quad (1)$$

Data preparation. The training set \mathbf{d} is composed of input labels summarized in Table 1 and output labels representing the shear stress components which will be applied as wall boundary conditions for wmLES. Before training the NN, feature selection is used to identify pairs of input/output that are highly correlated. Space-time correlations, illustrated in Appendix B (see Figure 5) are used to this end. When the flow separates, local correlations are insignificant and thus information has to be sought up- and down-stream. Therefore, it is necessary to enlarge the domain of dependence to include up- and down-stream information in both space and time. Figure 1 illustrates where data is extracted relative to the position along the wall at which the shear stress is predicted. Three wall-normal positions and 21 wall-parallel positions are considered. Currently, only instantaneous information is used for the training; future research will introduce time delays. The database is composed of $N_\xi \times N_z \times N_t$ pairs of input/outputs. Six instantaneous snapshots of the solution are interpolated on a $(N_\xi, N_\eta, N_z) = (180, 100, 10)$ probe grid per flow-through time t_c , defined as $t_c = L_x/u_b$. The database is created over $35t_c$, of which 90% is used for training, and the rest for testing. The validation database, used in Section 4, for *a priori* validation, covers $10t_c$ where snapshots are extracted at every wrLES time step, $(\Delta t u_b/h) = 5 \cdot 10^{-3}$.

Table 1: Inputs of the data-driven wall model (h is the hill height)

<i>Field</i>	<i>Description</i>	<i>Pre-scaling</i>
Velocity	\mathbf{u}	$u_\xi/h_{wm}, u_\eta/h_{wm}, u_z/h_{wm}$
Pressure Gradients	∇p	$(h_{wm}/h) (\partial p/\partial \xi), (h_{wm}/h) (\partial p/\partial \eta), (h_{wm}/h) (\partial p/\partial z)$
Length scale	h_{wm}	$\ln(h_{wm}/y^*)$

According to Yang et al. [4], it is essential to introduce some physical knowledge to the input features for a better generalization of the model. Following their guidelines and the work of Zhou et al. [5], the inputs are pre-scaled using y^* (see Table 1). The length scale y^* is a near-wall scaling compatible with the separation process inspired by Duprat et al. [12]. It is defined as, $y^* = \nu/u_{\tau,p}$ with $u_{\tau,p} = \sqrt{(u_\nu^2 + u_p^2)}$ where $u_\nu = \sqrt{(\nu u_\parallel/h_{wm})}$ and $u_p = |(\nu/\rho)\partial_x p|^{1/3}$; h_{wm} is the wall-normal distance at which the field is interpolated and fed to the model, and $u_\parallel = \sqrt{(u_\xi^2 + u_z^2)}$ is the norm of the wall-parallel velocity measured at that location.

Feedforward neural network. The present study uses two forked feedforward *multilayer perceptron (MLP)* neural networks MLP1 and MLP2 (Figure 6a), with the same structure, but trained on different data sets. MLP1 is trained over the reference database (DB1), while MLP2 uses 12

instantaneous snapshots over one t_c , instead of 6. This second enriched database (DB2) aims to better capture the intermittency. We also study the behavior of a *Gaussian Mixture Neural Network (GNN)* with $K = 2$ modes (Figure 6b), trained over DB2. These NN are further described in Appendix C. The *rectified linear unit (ReLU)* is selected as the activation function to avoid the vanishing gradient problem. After applying the physical scaling to each input/output pair, they are further normalized using a min-max scaling. Regarding the inputs, this scaling is applied to each field independently and not to each component individually to conserve the relative amplitude of the components. For the outputs, this scaling is applied per component because the spanwise wall shear stress is smaller than the streamwise one (a common scaling would lead to a wrong prediction of the smaller component). MLP1 and MLP2 are trained with the MSE loss function (Equation 1). The GNN is trained with a different loss (Equation 2), accounting for the mean and variance of each modes. The adam optimizer of Kingma and Ba [13], with its default parameters, is used to train the models. The training of MLP1 and MLP2 is stopped when the test loss starts to increase to avoid overfitting [14]. The NN models are created and trained with the open-source library PyTorch [15].

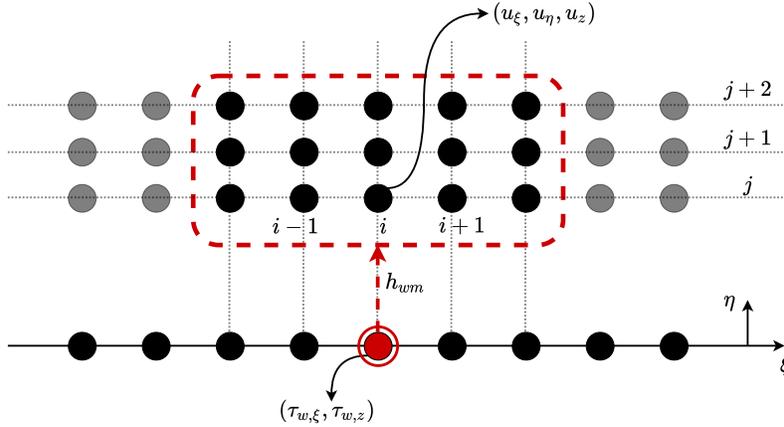


Figure 1: Generation of the database combining a single output (red dot circled) with a set of inputs (dotted red rectangle) spreading along the wall-normal and wall-parallel directions.

4 Preliminary results and a priori study

We first compare the distribution of the reference and predicted friction using boxplots (Figure 8) at three locations. The graphs confirm the observations of Figure 4. Near the separation (Figure 8a), the wide range of wall shear stress values is not captured by MLP1 and MLP2, while GNN recovers the mean and variance of the distribution. MLP1 and MLP2 are probably not able to retrieve the richness of the distribution due to the focus on regression of the mean in Equation 1. At the reattachment (Figure 8b), similar observations are made. However, the GNN slightly overestimates the variance. In the converging region (Figure 8c), the MLP1 captures better the mean, while the GNN is better at representing the variance; perhaps a third mode could improve its performance. Secondly, the mean shear stress is compared along the bottom wall; this average is computed over $10t_c$. As in Zhou et al. [5], we define its relative error as $\bar{\mathcal{E}} = (\hat{\bar{\tau}}_w - \bar{\tau}_w) / (|\bar{\tau}_w|_{\max})$, with $\bar{\cdot}$ the time averaging operator. Figure 2 shows the spatial evolution and the corresponding relative error. GNN captures the average separation well, although it does not understand the pre-separation, generating a relative error of 10% at $x/h \simeq 0$. MLP1 and MLP2 miss the mean separation line and directly assume that the flow separates at the hilltop. Regarding the recirculation bubble, there is an underestimation of the shear stress between $x/h \simeq 2$ and $x/h \simeq 3$. This inaccurate prediction is less important because the flow in the bubble does not directly drive the flow. The models capture the mean reattachment point well. Moreover, GNN correctly reproduces the reattachment between $x/h = 6$ and $x/h = 7$ with a relative error of less than 1%. The tiny separation at the hill foot is only predicted by MLP2. In the converging part, none capture the friction peak. MLP1 behaves best with only 25% relative error. GNN is the worst with 50% of relative error. Nonetheless, it predicts perfectly the wall shear stress up to $x/h = 8.5$ while MLP1 and MLP2 find a higher relative error at the windward face of the hill.

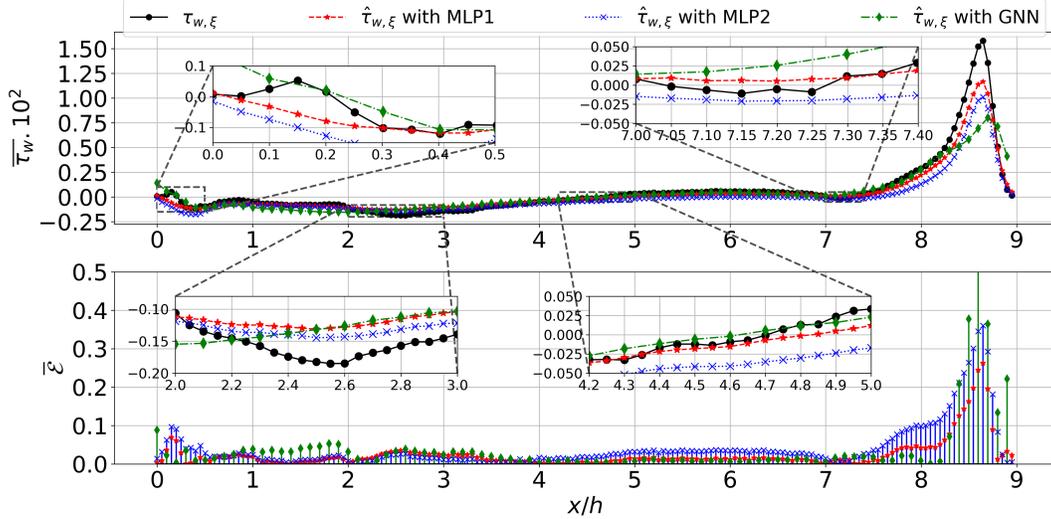


Figure 2: Comparison of the wall shear stress predicted by the data-driven wall models (MLP1, MLP2 and GNN) to the reference obtained by wrLES. Predicted values (top) and relative error (bottom).

5 Conclusion and future work

This work constitutes a first step towards a data-driven wall model for massively separated flows. The use of advanced correlations led to the selection of relevant input/output pairs and windowing information for the training of three DNNs. The *a priori* validation demonstrates satisfactory results for most of the bottom wall, between $x/h \simeq 1$ and $x/h \simeq 8$ for all models. GNN appears to correctly capture the mean separation while missing the friction peak. We expect that the use of more modes could improve this. We can furthermore question the spatial richness of the database, in particular at the hilltop where a quite complex flow is found. These points will be further investigated. We conclude that Gaussian Mixture Neural networks can be a great asset for the development of wall models applied to separation. In the future, we will include additional geometrical information such as the radius of curvature which is another parameter influencing separation. Furthermore *a priori* physical relations should be added to improve generalization. Our approach will include multiple streamwise and wall-normal positions. These positions are not yet encoded in the neural network, due to its architecture. Therefore, self-attention layers would be integrated as well.

Acknowledgments and Disclosure of Funding

The present research benefited from computational resources made available on the Tier-1 supercomputer of the Fédération Wallonie-Bruxelles, infrastructure funded by the Walloon Region under the grant agreement n°1117545. The funding of M. Boxho by SafranTech is gratefully acknowledged.

References

- [1] H. Choi and P. Moin. Grid-point requirements for Large Eddy Simulation: Chapman’s estimates revisited. *Phys. Fluids*, 24(1):011702, 2012.
- [2] Johan Larsson, Soshi Kawai, Julien Bodart, and Ivan Bermejo-Moreno. Large Eddy Simulation with modeled wall-stress: recent progress and future directions. *Mechanical Engineering Reviews*, 3(1):15–00418–15–00418, 2016.
- [3] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.
- [4] X. I. A. Yang, S. Zafar, J.-X. Wang, and H. Xiao. Predictive large-eddy-simulation wall modeling via physics-informed neural networks. *Phys. Rev. Fluids*, 4(3):034602, 2019-03-15.

ISSN 2469-990X. doi: 10.1103/PhysRevFluids.4.034602. URL <https://link.aps.org/doi/10.1103/PhysRevFluids.4.034602>.

- [5] Zhideng Zhou, Guowei He, and Xiaolei Yang. Wall model based on neural networks for LES of turbulent flows over periodic hills. *Phys. Rev. Fluids*, 6(5):054610, 2021. ISSN 2469-990X. doi: 10.1103/PhysRevFluids.6.054610. URL <https://link.aps.org/doi/10.1103/PhysRevFluids.6.054610>.
- [6] Adrián Lozano-Durán and Hyunji Jane Bae. Self-critical machine-learning wall-modeled LES for external aerodynamics. *arXiv:2012.10005 [physics]*, 2021. URL <http://arxiv.org/abs/2012.10005>.
- [7] Jochen Fröhlich, Christopher. P. Mellen, Wolfgang Rodi, Lionel Temmerman, and Michael A. Leschziner. Highly resolved large-eddy simulation of separated flow in a channel with stream-wise periodic constrictions. *J. Fluid Mech.*, 526:19–66, 2005.
- [8] Underlying Flow Regime case UFR 3-30: 2D Periodic Hill Flow, 2013. URL https://kbwiki.ercsoftac.org/w/index.php/Abstr:2D_Periodic_Hill_Flow.
- [9] C. Benocci and A. Pinelli. The role of the forcing term in the large-eddy simulation of equilibrium channel flow. *Engineering turbulence modeling and experiments*, page 287–96, 1990.
- [10] L. Temmerman, M. Leschziner, C. Mellen, and J. Fröhlich. Investigation of subgrid-scale models and wall-function approximations in large eddy simulation of separated flow in a channel with streamwise periodic constrictions. *Int J Heat Fluid Flow*, 24(2):157–180, 2003.
- [11] X. Gloerfelt and P. Cinnella. Investigation of the flow dynamics in a channel constricted by periodic hills. In *45th AIAA Fluid Dynamics Conference*. AIAA, 2015.
- [12] C. Duprat, G. Balarac, O. Métais, P. M. Congedo, and O. Brugière. A wall-layer model for large-eddy simulations of turbulent flows with/out pressure gradient. *Physics of Fluids*, 23(1):015101, 2011-01. ISSN 1070-6631, 1089-7666. doi: 10.1063/1.3529358. URL <http://aip.scitation.org/doi/10.1063/1.3529358>.
- [13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980 [cs]*, 2017. URL <http://arxiv.org/abs/1412.6980>.
- [14] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992. ISSN 0899-7667. doi: 10.1162/neco.1992.4.1.1. URL <https://www.mitpressjournals.org/doi/10.1162/neco.1992.4.1.1>.
- [15] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL <http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- [16] Dominic Edelmann, Tamás F. Móri, and Gábor J. Székely. On relationships between the Pearson and the distance correlation coefficients. *Statistics & Probability Letters*, 169:108960, 2021.
- [17] Gábor J. Székely, Maria L. Rizzo, and Nail K. Bakirov. Measuring and testing dependence by correlation of distances. *Ann. Statist.*, 35(6):2769–2794, 2007.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to **[Yes]**, **[No]**, or **[N/A]**. You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
 - (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? **[Yes]** The work scope is the prediction of separation and the enhancement of current wall models using DNN.
 - (b) Did you describe the limitations of your work? **[Yes]** see section 4
 - (c) Did you discuss any potential negative societal impacts of your work? **[No]** This work is to low TRL to discuss such societal impacts.
 - (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? **[Yes]** No potential negative societal impacts has been detected, this work has for future goal to better understand aero-engine and to be able to reduce their carbon impact.
2. If you are including theoretical results...
 - (a) Did you state the full set of assumptions of all theoretical results? **[N/A]** Not applicable here.
 - (b) Did you include complete proofs of all theoretical results? **[N/A]** Not applicable here.
3. If you ran experiments...
 - (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? **[Yes]** See section 2
 - (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? **[Yes]** See section C
 - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? **[No]** No, we run numerical simulation whose averaged results compared well with the literature.
 - (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? **[Yes]** See section 3
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
 - (a) If your work uses existing assets, did you cite the creators? **[Yes]** Data are generated with Argo-DG developed at Cenaero (see section 2)
 - (b) Did you mention the license of the assets? **[Yes]** PyTorch is used as an open-source library and is cited in the reference [15]
 - (c) Did you include any new assets either in the supplemental material or as a URL? **[N/A]**
 - (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? **[No]** The database is not linked to "people".
 - (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? **[No]** The database does not contain such feature, we only deal with instantaneous value of the wall shear stress.
5. If you used crowdsourcing or conducted research with human subjects...
 - (a) Did you include the full text of instructions given to participants and screenshots, if applicable? **[N/A]** Not applicable here.
 - (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? **[N/A]** Not applicable here.
 - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? **[N/A]** Not applicable here.

A APPENDIX: VALIDATION

Our results obtained on the two-dimensional periodic hill at $Re_b = 10595$ are compared with those obtained by Temmerman et al. [10] and by Gloerfelt and Cinnella [11]. Figure 3 shows the vertical profiles of the time-averaged streamwise velocity \bar{u}/u_b and the Reynolds shear stress $\overline{u'u'}/u_b^2$. Considering the average velocity profiles and Reynolds stresses, good agreement with the references is observed.

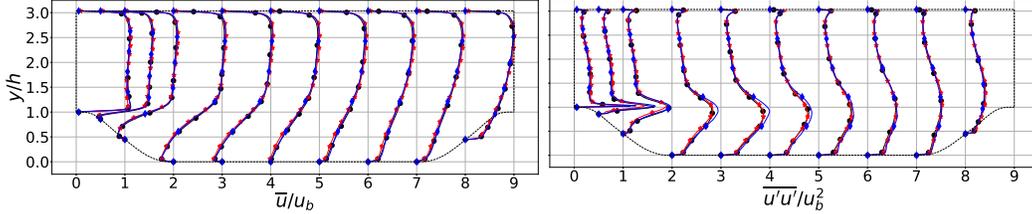


Figure 3: Averaged velocity profile \bar{u}/u_b (top) and stress component $\overline{u'u'}/u_b^2$ (bottom) of the 2D Periodic Hill flow at $Re_b = 10595$, compared with ERCOFTAC database (star red curve), X. Gloerfelt and P. Cinnella at $M = 0.1$ (diamond blue curve).

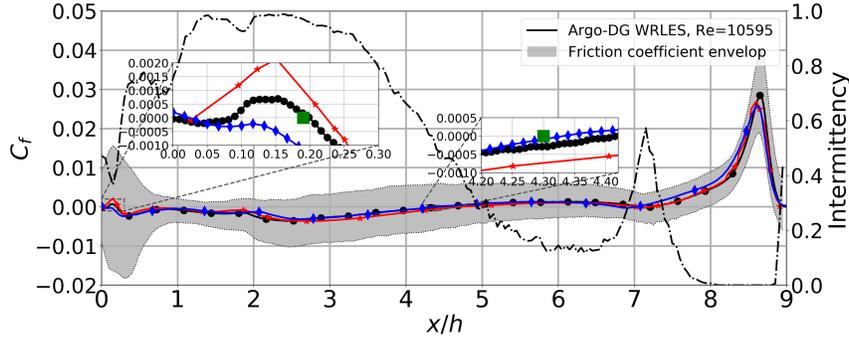


Figure 4: Friction coefficient C_f measured on the bottom surface of the periodic hill (circle black curve), compared with ERCOFTAC database (star red curve), X. Gloerfelt and P. Cinnella at $M = 0.1$ (diamond blue curve). The graph depicts also the friction coefficient envelop (gray area), intermittency γ measurement (dash-dotted line) and mean separation and reattachment locations (green squares).

The friction coefficient defined as $u_\tau^2/(0.5\rho u_b^2)$ where $u_\tau = \sqrt{\tau_w/\rho_w}$ is the friction velocity, is shown in Figure 4. There is good agreement with the results obtained by Gloerfelt and Cinnella [11] at $M = 0.1$. A precursory separation is visible at $x/h \simeq 0$. According to Gloerfelt and Cinnella [11], this pre-separation arises only when using compressible flow solvers. The mean separation line is located at $x/h \simeq 0.19$. In the separation bubble extending to $x/h \simeq 4.21$, the flow decelerates and then re-accelerates due to the change in curvature. We note two divergences: (i) the deceleration/acceleration process is slightly shifted downstream, and (ii) the secondary separation bubble at the bottom of the hill is predicted to occur before $x/h = 7.0$. We suspect an alternative definition of the geometry and, more specifically, at the junction between the flat part of the bottom and the hill. Our DG solution captures well the mean reattachment location. The recovery region extends from $x/h \simeq 4.2$ to $x/h \simeq 7.0$, with a secondary recirculation bubble at $x/h \simeq 7.2$, followed by a strong acceleration on ascent.

B APPENDIX: CORRELATION MAPS

Before training any machine learning models, it is important to analyse the database to select relevant input/output pairs. Feature selection can improve model performances and reduces the computational cost of modeling. Space-time correlations (Pearson [16] and distance proposed by Székely et al. [17]) are adopted to perform feature selection for the present problem. While Pearson correlations only detect linear relations between data, distance correlations can extract both linear and non-linear

relationships. An example of such correlation map is shown on Figure 5. These correlation maps expose the relation between the instantaneous wall shear stress $\tau_{w,\xi}$ and the instantaneous wall-parallel velocity component u_ξ . The main observation is the domain of dependence shift in the positive $\delta\xi/h$ with a maximum value of 0.45 (for Pearson correlation). We conclude that separation drives the behavior of the wall shear stress at $x/h \simeq 0.05$.

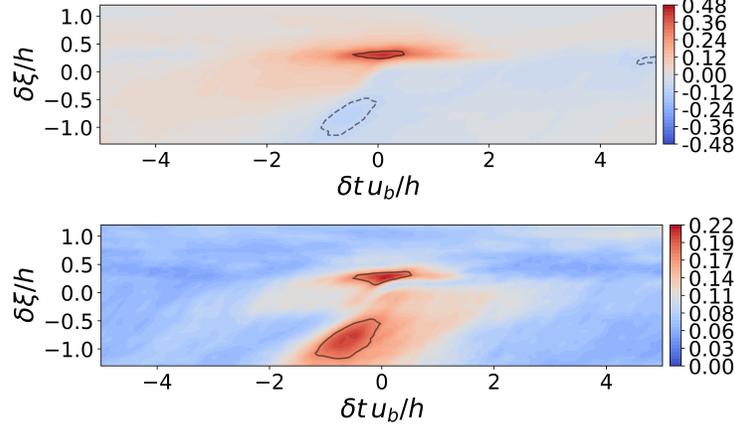


Figure 5: Pearson (top) and distance correlation (bottom) between $\tau_{w,\xi}$ and u_ξ at $(\xi, \eta)/h \simeq (0, 0.1)$.

C APPENDIX: NEURAL NETWORK ARCHITECTURE

Description of MLP1 and MLP2. The neural network used for the present study is a forked fully-connected feedforward neural network with a common set of parameters (blue rectangle) and a set of target parameters (red rectangles), dedicated to each task. There are two distinct tasks (green rectangle): (1) the prediction of the streamwise wall shear stress $\tau_{w,\xi}$ and (2) the prediction of the spanwise wall shear stress $\tau_{w,z}$. The common neural network is composed of seven hidden layers with 400, 200, 200, 200, 100, 100, 50 neurons. Each target neural network is composed of one hidden layer of 50 neurons. The inputs are clearly described in Table 1. The training is stopped once the test loss (gray curve on Figure 7) starts to increase to avoid overfitting. The neural network model is thus saved after 150 epochs (red star). Moreover, after 150 epochs, the success rate measured based on the relative error is above 98% for both outputs.

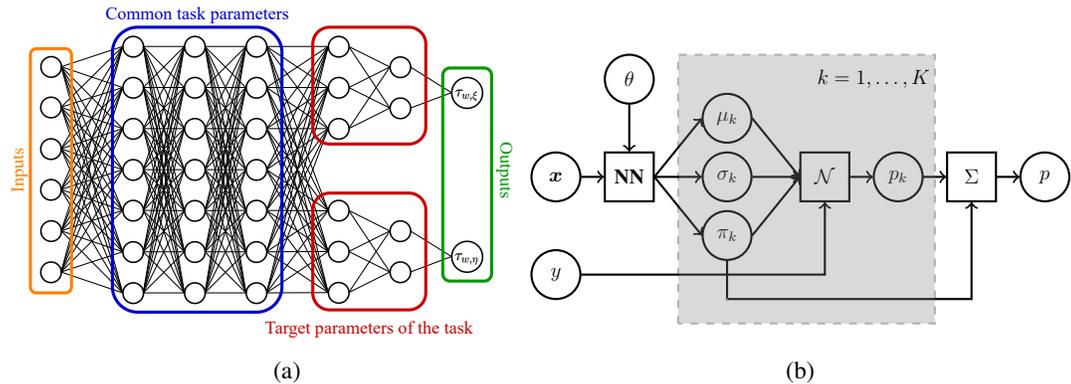


Figure 6: Schematic of the architecture of (a) a feedforward neural network with common parameters and target parameters for the prediction of the wall shear stress components; (b) a Gaussian Mixture Neural Network with K modes.

Description of GNN. This NN is depicted in Figure 6b where the distribution of $\tau_{w,\xi}$ is predicted. This NN is trained via a different loss function accounting for the mean and variance of the K modes.

For the present study, only two modes are considered ($K = 2$). The loss function is:

$$\mathcal{L} = \sum_{\mathbf{x}_i, y_i \in \mathcal{d}} \log \left(\sum_{k=1}^K \frac{1}{\sqrt{2\pi}\sigma_k(\mathbf{x}_i)} \exp \left(-\frac{(y_i - \mu_k(\mathbf{x}_i))^2}{2\sigma_k^2(\mathbf{x}_i)} \right) \right), \quad (2)$$

where μ_k , σ_k and π_k are respectively the mean, the variance and the mixture coefficient of mode k . The network contains four hidden layers with respectively 400, 200, 100, and 20 neurons each. The inputs are those described in Table 1 while it tries to predict the distribution of $\tau_{w,\xi}$ only. $\tau_{w,z}$ distribution is not inferred here. The model is evaluated after 100 epochs (Figure 7).

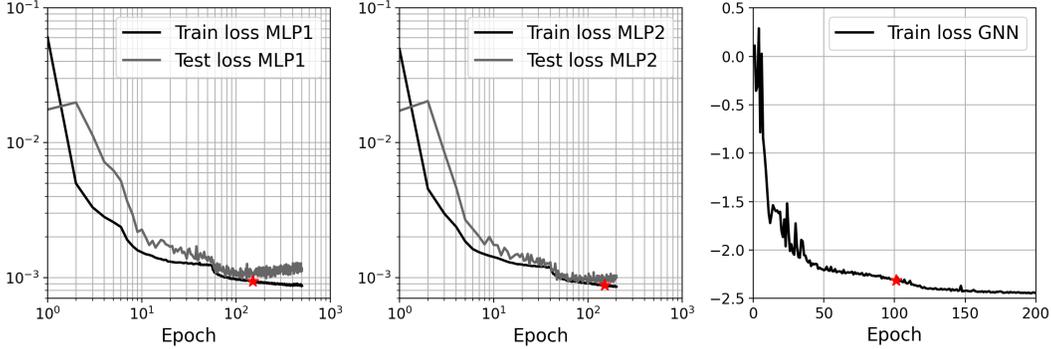


Figure 7: Evolution of the losses with training epochs for both the training and testing datasets: MLP1 (left), MLP2 (middle) and GNN (right) with red stars being the epoch at which models are evaluated.

Section 4 discusses the validation of the model prediction against wrLES results in an *a priori* study. As a first step, model predictions are plotted in boxplot (see Figure 8) to have a more in-depth view of their distribution. The analysis of Figure 8 is already done in the results section.

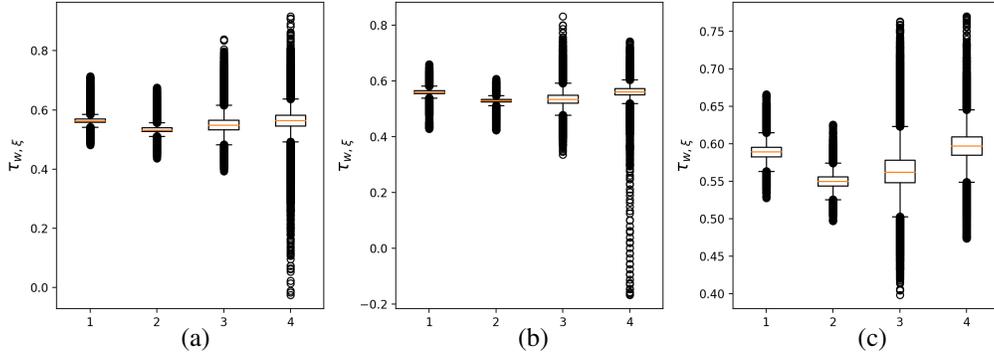


Figure 8: Boxplot comparing the model prediction (1:MLP1, 2:MLP2 and 3:GNN) and the ground truth distribution (4) for three different locations along the wall: (a) $\xi/h \approx 0$, (b) $\xi/h \approx 4.5$ and (c) $\xi/h \approx 8$.