Contents

1	Introduction	1
2	Related Work	2
3	Dataset Presentation	3
4	Benchmarking Setup	5
5	Benchmarking Results	7
6	Conclusion	10
7	Paper Checklist	16
A	Resources Availability and Licensing	18
B	Broader impact	18
C	Reproducibility statement	18
D	Description Of Software	18
E	Constant and Dimensionless Quantities	19
F	Reynolds-Averaged Navier–Stokes Equations	20
G	Force Coefficients	22
H	Airfoil Generation and Statistics	22
I	Meshing Procedure	25
J	Boundary Conditions	26
K	Simulation Validation	28
L	Models architecture	31
M	Additional Results	33

17

A Resources Availability and Licensing

This work is open-source under Open Data Commons Open Database License v1.0. For both dataset generation (mesh generation, OpenFoam simulations) and ML experiments/baselines. The Open Database License (ODbL) is a license agreement that allows users to freely share, modify, and use a database while maintaining this same freedom for others. The **Permissions** include: • Commercial use • Distribution • Modification • Private use . The **limitations** are: • Liability • Patent use • Trademark use • Warranty and finally the **conditions** are: • Disclose source f • License and copyright notice • Same license. This license is the same than the one used in our first version of this work [4].

B Broader impact

This work could be used to:

1. experiment new Geometric Deep Learning (GDL) models in this area,
2. study the capabilities of Deep Learning (DL) to capture physical phenomena relying on our physics-based evaluation protocol,
3. give insights to establish new research directions for combining numerical simulation and Machine Learning (ML) following the behaviors that will be observed in our quantitative and qualitative results w.r.t the targeted physical metrics that we developed in purpose for this work,
4. extend graph-mesh, and point clouds benchmarking datasets and applications of GNNs to real-world-like physics problems,
5. build surrogate solvers to help Computational Fluid Dynamics (CFD) engineers optimize design cycles and iterate as efficiently as needed on their designs,
6. reduce the cost (time and materials) of prototyping new designs of planes while avoiding dangerous studies in the real world, as well as enabling the test of several configurations,
7. study the generalization/extrapolation abilities of DL to large/unseen domains as the industrial world demands, including wide ranges of initial and boundary conditions.

C Reproducibility statement

We provide a GitHub repository to reproduce all the experiments and a link to download the preprocessed dataset and another one for the raw OpenFOAM data. The experiments have been conducted with a single NVIDIA RTX 3090 24Go. The repository the preprocessed/raw datasets include: • code to reproduce the ML experiments • code to generate the figures.

We also provide a GitHub repository to run new simulations and to be able to reproduce the generation settings of the dataset. The simulations have been done with 16 CPU cores of an AMD Ryzen™ Threadripper™ 3960X. The codes in the repository include: • (extensible) code to generate the meshes • code to run new simulations and/or build the dataset.

Finally, we provide the AirfRANS Python library with its associated GitHub repository to easily manipulate simulations from the dataset.

D Description Of Software

In this section, we describe the tools that we have used in this work to build the dataset[5], make the visualizations, and train the models. This work makes use of computational fluid dynamics (CFD) and ML tools.

OpenFOAM [64] stands for *Open-source Field Operation And Manipulation*, a C++ software for developing custom numerical solvers to study continuum mechanics and CFD problems. In this work, we have used version v2112 of OpenFOAM to make our simulations. OpenFOAM is released as free and open-source software under the *GNU General Public Licence*.

---

[5]similar to our first version of our dataset [4]

18

Table 6: Properties of air at 298.15 K (25 °C) and at sea level on earth.

Name	Symbol	Value
Kinematic viscosity	ν	$1.56 \times 10^{-5}\,\mathrm{m^2\,s^{-1}}$
Specific mass*	ρ	$1.184\,\mathrm{kg\,m^{-3}}$
Thermal diffusivity*	α	$2.25 \times 10^{-5}\,\mathrm{m\,s^{-1}}$
Specific heat*	c_p	$1005\,\mathrm{J\,kg^{-1}\,K^{-1}}$
Atmospheric pressure*	p_0	$1.013 \times 10^5\,\mathrm{N\,m^{-2}}$
Atmospheric temperature*	T_0	$298.15\,\mathrm{K}$
Speed of sound in the fluid*	c	$346.1\,\mathrm{m\,s^{-1}}$

* Those values are important only in the compressible case, they are given for comparison with compressible simulations. Especially, the absolute pressure is set to $0\,\mathrm{N\,m^{-2}}$ for the incompressible case as it only depends on the differential pressure.

ParaView [3] is an open-source visualization tool designed to explore and visualize efficiently large data using quantitative and qualitative metrics. ParaView runs on distributed and shared memory parallel and single processor systems. In this work, we have used it to visualize the following: point clouds, meshes, the predicted (as well as the ground truth) physical fields. We have used version 5.10.0 of ParaView in this work. ParaView is released as free and open-source software under the *Berkeley Software Distribution License*.

PyVista [57] is an open-source tool based on a handy interface for the Visualization ToolKit (VTK). It is simple to use in interaction with NumPy [28] and other Python libraries. It is mainly used for mesh analysis. In this work, we use PyVista to build the inputs of our DL models. We have used version 0.36.1 of PyVista in this work. PyVista is released as free and open-source software under the *MIT License*.

PyTorch [46] is an open-source library for DL using GPUs and CPUs. In this work, we use PyTorch to build our training protocol. In this work, we have used version 1.11.0 of Pytorch along with CUDA 11.3 and Python 3.9.12. PyTorch is released as free and open-source *Berkeley Software Distribution License*.

PyTorch Geometric (PyG) [19] is an open-source library for GDL built upon PyTorch which targets the training of geometric neural networks, including point clouds, graphs and meshes. We use PyG to design our message passing schemes. In this work, we have used version 2.0.4 of PyG along with CUDA 11.3. PyG is released as free and open-source software under the *MIT License*.

E Constant and Dimensionless Quantities

The fluid used in this study is the air at 298.15 K (25 °C) and at sea level on earth. In Table 6 we give the different values of the constant associated with this fluid.

The only dimensionless quantity for the incompressible case is the Reynolds number Re. We add the Mach number Ma and the Prandtl number Pr in the compressible case. Those quantities are defined by:

$$Re = \frac{UL}{\nu}, \quad Ma = \frac{U}{c}, \quad Pr = \frac{\nu}{\alpha} \tag{4}$$

where U is the characteristic velocity of the problem, L its characteristic length, c the speed of sound in the fluid, ν its kinematic viscosity and α its thermal diffusivity. The Reynolds number compares the order of magnitude of the convective term with respect to the diffusive term in the Navier–Stokes equations, the Mach number quantifies flow compressibility and the Prandtl number compares the order of magnitude of the variation of energy via momentum with respect to the variation of energy via heat transfer in the compressible form of Navier–Stokes equations.

19

F Reynolds-Averaged Navier–Stokes Equations

Under certain assumptions, the dynamics of a fluid is governed by the Navier–Stokes equations. Those equations are composed of a continuity equation, three momentum equations and an energy equation (see §49 of [34] and §5.3 of [65]):

$$\partial_t \rho + \partial_i(\rho u_i) = 0 \tag{5}$$

$$\partial_t(\rho u_i) + \partial_j(\rho u_j u_i) = -\partial_i p + \partial_j \sigma_{ji}, \quad i \in \{1,2,3\} \tag{6}$$

$$\partial_t \left(\rho \left(\epsilon + \frac{1}{2} u^2 \right) \right) + \partial_j \left(\rho u_j \left(h + \frac{1}{2} u^2 \right) \right) = \partial_j(u_i \sigma_{ij}) - \partial_j q_j \tag{7}$$

where ∂_t denotes a partial derivative with respect to time, ∂_i a partial derivative with respect to the ith space coordinate, ρ the fluid specific mass, u the fluid velocity, p the fluid pressure, σ the viscous stress tensor, ϵ the fluid specific energy, h the fluid specific enthalpy ($h := \epsilon + p/\rho$) and q the heat flux density due to thermal conduction. Moreover, we used the Einstein summation convention over repeated indices. Finally, fluid dynamics theory, thermodynamic relations, Fourier law and the perfect gas law give us:

$$\sigma_{ij} = \mu \left(\partial_i u_j + \partial_j u_i - \frac{2}{3} \partial_k u_k \delta_{ij} \right), \quad i,j \in \{1,2,3\} \tag{8}$$

$$\epsilon = c_v T \tag{9}$$

$$h = c_p T \tag{10}$$

$$q_i = -\kappa \partial_i T, \quad i \in \{1,2,3\} \tag{11}$$

$$p = \rho R T \tag{12}$$

where δ_{ij} is the kronecker tensor, T the fluid temperature, μ the fluid dynamic viscosity (function of T), κ the fluid thermal conductivity (function of T), R the fluid specific constant, c_v and c_p the fluid specific heat coefficient for constant volume and pressure respectively (taken constant here). This leads to a close set of partial differential equations with 6 unknowns (ρ, p, u, T) and 6 equations:

$$\partial_t \rho + \partial_i(\rho u_i) = 0 \tag{13}$$

$$\partial_t(\rho u_i) + \partial_j(\rho u_j u_i) = -\partial_i p + \partial_j \left(\mu \left(\partial_i u_j + \partial_j u_i - \frac{2}{3} \mu \partial_k u_k \delta_{ij} \right) \right), \quad i \in \{1,2,3\} \tag{14}$$

$$\partial_t \left(\rho \left(c_v T + \frac{1}{2} u^2 \right) \right) + \partial_j \left(\rho u_j \left(c_p T + \frac{1}{2} u^2 \right) \right) = \partial_j \left(\mu u_i \left(\partial_i u_j + \partial_j u_i - \frac{2}{3} \partial_k u_k \delta_{ij} \right) \right) + \partial_j(\kappa \partial_j T) \tag{15}$$

together with the state equation 12. We chose the perfect gas law for the state equation as we are going to treat the case of air but this equation can be replaced by any state equation better suited for the problem.

In certain cases, we can decently assume that the fluid is incompressible with constant density ρ. We then need only 4 equations to close the problem. We get rid of the energy and state equations and find the incompressible Navier–Stokes equations:

$$\partial_i u_i = 0 \tag{16}$$

$$\partial_t u_i + \partial_j(u_i u_j) = -\partial_i \left(\frac{p}{\rho} \right) + \nu \partial_{jj}^2 u_i, \quad i \in \{1,2,3\} \tag{17}$$

where $\nu := \mu/\rho$ is the fluid kinematic viscosity, taken constant in this case. In order to explicitly write an important dimensionless quantity in fluid mechanics, we can rewrite last equations with dimensionless variables. Let us define, T, U, L and P characteristic time, velocity, length and pressure of the problem, respectively. We write:

$$t = T\hat{t}, \quad u = U\hat{u}, \quad x = L\hat{x}, \quad p = P\hat{p} \tag{18}$$

where x is the cartesian position, $\hat{t}$, $\hat{u}$, $\hat{x}$ and $\hat{p}$ are dimensionless quantities. If we write $P = \rho U^2$ and $T = L/U$, we find for the incompressible case:

$$\partial_{\hat{i}} \hat{u}_i = 0 \tag{19}$$

$$\partial_{\hat{t}} \hat{u}_i + \partial_{\hat{j}}(\hat{u}_i \hat{u}_j) = -\partial_{\hat{i}} \hat{p} + \frac{1}{Re} \partial_{\hat{j}\hat{j}}^2 \hat{u}_i, \quad i \in \{1,2,3\} \tag{20}$$

20

where $Re := UL/\nu$ is the Reynolds number. This dimensionless number quantifies the importance of the convective term with respect to the diffusive term (in order of magnitude):

$$\frac{\|\partial_j(u_i u_j)\|}{\nu\|\partial_{jj}^2 u_i\|} \approx \frac{UL}{\nu} = Re \tag{21}$$

When the Reynolds number tends to 0, diffusion term are dominant, we call it a Stokes flow. When the Reynolds number tends to ∞, the equations get closer to the Euler equations for inviscid fluid. At high Reynolds, new chaotic patterns can emerge close to walls and the different fields get untidy. This transition is the transition between what we call laminar (tidy) and turbulent (untidy) flows. Turbulence is a process that emerges at high Reynolds number and allows more dissipation than expected with laminar flows via the emergence of eddies of different length scales (see §33 of [34]). Theoretical resolution of such dynamics is an open problem and direct numerical simulations (DNS) are highly challenging because of their huge computational costs. Hence, different technologies have been developed in order to reduce the computational complexity of the task, for example, large eddy simulations (LES) try to filter in space the pressure and velocity fields and model the smallest eddies by adding dissipation. Another one, that we will use here, try to model all the scales of eddies by doing an ensemble average on the velocity and pressure fields. An ensemble average is a theoretical average over multiple equivalent experiments, this can also be equivalently replaced by a time averaging on a time scale big compared to turbulent fluctuations rate and small compared to the macroscopic evolution rate of the fluid. We write:

$$u = \bar{u} + u', \quad p = \bar{p} + p' \tag{22}$$

where $\bar{\cdot}$ denotes an ensemble averaged quantity and $\cdot'$ its fluctuations. If we set those expressions into the incompressible Navier–Stokes equations and take the ensemble averaging of the equations, we get:

$$\partial_i \bar{u}_i = 0 \tag{23}$$

$$\partial_t \bar{u}_i + \partial_j(\bar{u}_i \bar{u}_j) = -\partial_i\left(\frac{\bar{p}}{\rho}\right) + \nu\partial_{jj}^2 \bar{u}_i - \frac{1}{\rho}\partial_j(\sigma_t)_{ij}, \quad i \in \{1,2,3\} \tag{24}$$

where $(\sigma_t)_{ij} := -\rho\overline{u_i' u_j'}$ is called the Reynolds stress tensor. We now have a new unknown in our equations and the problem is not close anymore. A common assumption known as the Boussinesq hypothesis is to write the Reynolds stress tensor in the same way as the viscous stress tensor:

$$(\sigma_t)_{ij} = \rho\nu_t \left(\partial_i \bar{u}_j + \partial_j \bar{u}_i\right) - \frac{2}{3}\rho k \tag{25}$$

$$k = \frac{1}{2}\overline{u_i' u_i'} \tag{26}$$

where ν_t is called the turbulent kinematic viscosity and k the specific kinematic energy of turbulence. The term $-2\rho k/3$ is set to ensure the null value of the trace of σ_t. By defining an effective pressure, abusively denoted by the same symbol, $\bar{p}$, $\bar{p} := \bar{p} + 2\rho k$, we find:

$$\partial_i \bar{u}_i = 0 \tag{27}$$

$$\partial_t \bar{u}_i + \partial_j(\bar{u}_i \bar{u}_j) = -\partial_i\left(\frac{\bar{p}}{\rho}\right) + \partial_j\left[(\nu + \nu_t)\partial_j \bar{u}_i\right], \quad i \in \{1,2,3\} \tag{28}$$

This set of equations is known as the Reynolds-Averaged Navier–Stokes (RANS) equations. In order to close our set of equations, we need a last equation for ν_t. Such equation is called a turbulence model and plenty of them have been developed in the last decades to recover experimental results in certain environments. We very briefly present two turbulence models that we are going to use in our experiments and let the details of those model in the references given. The Spalart-Allmaras model [56] is a one-equation model designed for aerodynamics problems, it involves a modified viscosity called $\tilde{\nu}$. The $k - \omega$ SST model [42] is the blending of two two-equations turbulence model, namely the $k - \varepsilon$ and the $k - \omega$ models [35, 66], and it extends the domain of application of both by switching models where it is more relevant to use one instead of another. It involves two quantities, the specific kinematic turbulent energy k and the specific turbulence dissipation rate ω.

In the compressible case, a mass-average is applied to the Navier–Stokes equations, for example in the case of the velocity, we use:

$$\tilde{u} = \frac{1}{\bar{\rho}}\overline{\rho u} \tag{29}$$

21

and we can decompose the velocity in a mass-averaged term and a fluctuation term:

$$u = \tilde{u} + u'' \tag{30}$$

By doing this for different quantities such as specific energy, specific enthalpy, temperature etc... we can write a new set of equations in a similar form as 5 - 7. This is called the Favre-Averaged Navier–Stokes equations, details can be found in chapter 5 of [65].

Finally, the RANS equations are the equations solved by the *simpleFoam* solver and the Favre-Averaged Navier–Stokes equations the ones solved by the *rhoSimpleFoam* solver in the OpenFOAM suite. We compare our results with compressible simulations and the results given in the TMR [10] in Appendix K.

G Force Coefficients

The stress force df acting on a face of area dS and normal n is:

$$df = -pn + 2\mu S \cdot n \tag{31}$$

We can conclude that for a geometry of surface $\mathcal{S}$, the stress force F acting on it can be computed via:

$$F = \oint_{\mathcal{S}} \sigma \cdot n dS \tag{32}$$

$$= -\oint_{\mathcal{S}} pn dS + \oint_{\mathcal{S}} 2\mu S \cdot n dS \tag{33}$$

We call the term $P := -pn$ the wall pressure and the term $\tau := 2\mu S \cdot n$ the wall shear stress. Ultimately, we call drag D and lift L the component of F that are respectively parallel and orthogonal to the main direction of the flow. If $u_\parallel$ is the unit direction of the velocity vector and $u_\perp$ its orthogonal unit direction, we have:

$$D = \left(\oint_{\mathcal{S}} p dS + \oint_{\mathcal{S}} \tau dS \right) \cdot u_\parallel \tag{34}$$

$$L = \left(\oint_{\mathcal{S}} p dS + \oint_{\mathcal{S}} \tau dS \right) \cdot u_\perp \tag{35}$$

In the case of RANS equations, we add terms that take in account the effect of turbulence over the geometry. The pressure p is replaced by an effective mean-field pressure $\bar{p}$ and the wall shear stress is given by $\tau = 2(\mu + \mu_t)S \cdot n$ where μ_t is the dynamic turbulent viscosity. However, as the turbulent viscosity is null over the airfoil, we recover $\tau = 2\mu S \cdot n$.

For incompressible fluids we also often divide those quantities by ρ the density of the fluid and solvers often express the results in terms of reduced pressure $\bar{p} \rightarrow \bar{p}/\rho$ and kinematic (turbulent) viscosity $\nu := \mu/\rho$ ($\nu_t := \mu_t/\rho$). We use this convention in this work.

H Airfoil Generation and Statistics

In this section, we review the construction of the NACA 4 and 5 digits [14]. Both of them are built in the same manner and rely only on 3 or 4 parameters for the 4 or 5 digits respectively. Each airfoil is defined via a camber lined and an envelope, the only difference between the 4 and 5 digits is the definition of the camber line.

NACA 4 digits.   Those profiles are defined by the name NACA followed by four digits MPXX where the first two digits M and P defined the camber line and the last two digits XX defined the maximum thickness of the profile in percentage of the chord (the total length of the airfoil). More precisely, M defines the maximum ordinate of the camber line in hundredth of the chord and P the position of this maximum from the leading edge in tenth of the chord. If we denote the chord c, the camber line of the NACA 4312 profile will have a maximum ordinate of camber of $y = 0.04c$, at $x = 0.3c$ and the profile will have a maximum thickness of $0.12c$. Also, the leading edge and the trailing edge of each airfoil are always taken at the points $(0, 0)$ and $(c, 0)$ in the $x - y$ plane respectively. From this point, all the abscissas and ordinates will be given in length per chord.

22

For the NACA 00XX, a symmetrical profile, the camber line is a straight line from $x = 0$ to $x = 1$ and the upper surface is defined by the graph of the function:

$$y_t(x) = \frac{t}{0.2} \left(0.2969\sqrt{x} - 0.126x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4 \right) \tag{36}$$

where $t := XX/100$ is the thickness defined with the two last digits. This definition involve a trailing edge with a thickness of $0.002c$. If, for example for numerical propose, we want to have a thickness of 0 at the trailing edge, we can change the coefficient of the fourth order term from -0.1015 to -0.1036. The lower surface is defined as $-y_t$.

For a generic NACA MPXX, The camber line is defined by the first two digits and follow the graph of the function:

$$y_c(x) = \begin{cases} m\frac{x}{p^2}(2p - x), & 0 \leqslant x \leqslant p \\ m\frac{1-x}{(1-p)^2}(1 + x - 2p), & p < x \leqslant 1 \end{cases} \tag{37}$$

where $m := 0.01\text{M}$ and $p := 0.1\text{P}$.

Finally, the upper surface of a generic NACA MPXX is given by the set of coordinates (x_u, y_u) defined as:

$$\begin{cases} x_u = x - y_t(x)\sin\theta(x) \\ y_u = y_c(x) + y_t\cos\theta(x) \qquad \text{for } x \in [0, 1] \\ \theta(x) = \arctan y_c'(x) \end{cases} \tag{38}$$

where y_c' is the derivative of y_c. The lower surface is given by a similar set of coordinates (x_l, y_l) defined as:

$$\begin{cases} x_l = x + y_t(x)\sin\theta(x) \\ y_u = y_c(x) - y_t\cos\theta(x) \qquad \text{for } x \in [0, 1] \\ \theta(x) = \arctan y_c'(x) \end{cases} \tag{39}$$

NACA 5 digits. As we already stated earlier, the only difference between the NACA 4 and 5 digits is the definition of the camber line. In the case of the 5-digits LPQXX, the 3 first parameters defining the camber line are less explicit than for the 4 digits case but allow more complex shapes. The last two are the same as in the 4-digits case (*i.e.* maximum thickness in hundredth of chord). The first digit L controls the camber implicitly via an optimal lift coefficient C_L, *i.e.* it will give you the camber of the airfoil such that $C_L = 0.15\text{L}$. The second digit P is almost the same as in the 4-digits case, *i.e.* it defines the position of the maximum of camber of the camber line in twentieth of chord. Lastly, the third digit Q is either 0 or 1 and represent a standard camber (similar to the 4-digits case) for 0 and a reflex camber for 1. The reflex camber is a double cambered line that makes the profile more stable (by setting the pitch coefficient to 0).

For a generic NACA LPQXX, the standard camber line is defined via the graph of the function:

$$y_c(x) = \begin{cases} K_1 \left(m^2(3 - m)x - 3mx^2 + x^3 \right), & 0 \leqslant m \\ K_1 m^3(1 - x), & m < x \leqslant 1 \end{cases} \tag{40}$$

where m is not the position of the maximum camber $p := 0.05\text{P}$ but is related to it via the equation:

$$p = m \left(1 - \sqrt{\frac{m}{3}} \right) \tag{41}$$

and K_1 is related to C_L via:

$$\begin{cases} K_1 = \frac{C_L}{Q} \\ Q = \frac{3m - 7m^2 + 8m^3 - 4m^4}{\sqrt{m(1-m)}} - \frac{3}{2}(1 - 2m)\left(\frac{\pi}{2} - \arcsin(1 - 2m)\right) \end{cases} \tag{42}$$

The reflex camber line is defined via the graph of the function:

$$y_c(x) = \begin{cases} K_1 \left(m^3(1 - x) - K_2(1 - m)^3 x + (x - m)^3 \right), & 0 \leqslant m \\ K_1 \left(m^3(1 - x) - K_2(1 - m)^3 x + K_2(x - m)^3 \right), & m < x \leqslant 1 \end{cases} \tag{43}$$

where m and K_1 are defined as in the standard case and K_2 is defined via:

$$K_2 = \frac{3(m - p)^2 - m^3}{(1 - m)^3} \tag{44}$$

We use a standard Newton's method to numerically solve equation 41 in m.

23
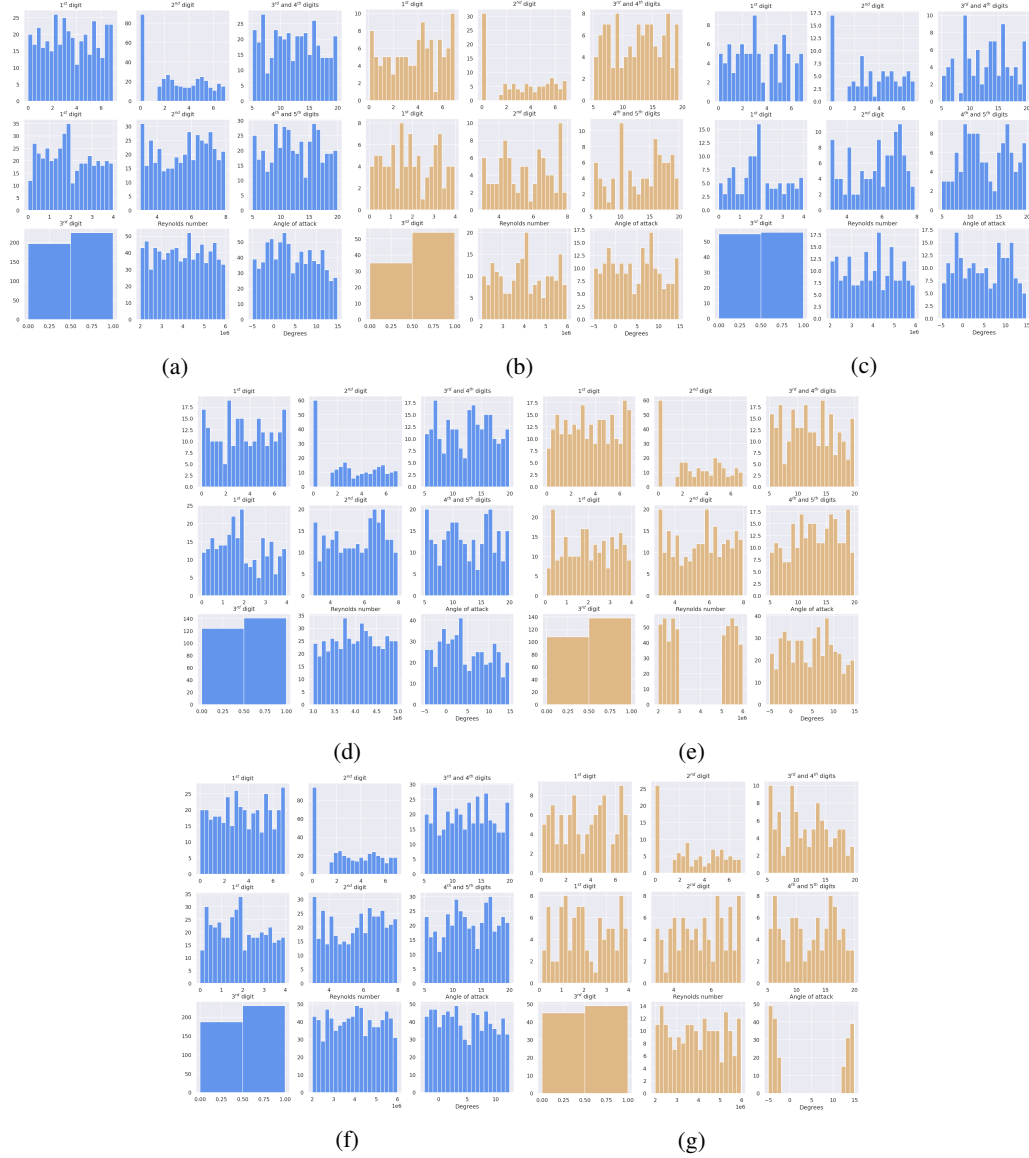
Figure 4: Histogram of the different sampled parameters for the airfoils. For each subfigure, the top line represents the paramters of NACA 4-digits, the middle line, the parameters of the NACA 5-digits along with the left plot of the bottom line. The last two plots of the bottom lines are the histograms for the Reynolds number and the angle of attack. Each subfigure represents a different regime: (a)-(b) Full data regime (c) Scarce data regime (d)-(e) Reynolds extrapolation regime (e)-(f) Angle of attack extrapolation regime. Trainsets are in blue and testsets in yellow.

Parameter sets. In Table 1 we give the parameter sets for the sampling. Let us underline that the digits are not necessarily integers. Also, for the values of P in the NACA 4-digits case, we actually uniformly sample in $[0, 7]$ and set the values of P strictly inferior to 1.5 to 0. This implies that the NACA 4-digits set is slightly biased towards symmetrical profiles. We assume that this bias is not of a great importance in the machine learning task. This bias could actually be leveraged to produce a smaller dataset of only symmetrical profiles where we can test the performance of invariant/equivariant models with respect to the plane symmetry of axis $y = 0$. Finally, we set the chord c to $1\,\mathrm{m}$ for all of the airfoils. In Figure 4, we show statistics of the airfoil parameters.

24

I   Meshing Procedure

The construction of meshes is at the core of CFD tasks. A mesh completely determine the quality of a simulation and its characteristics. For CFD problems, the local size of cells gives the information of the resolved scales. For example, in turbulent regime, it would be impossible to simulate eddies of characteristic length smaller than your typical cell length scale. Unfortunately, it is often impossible to run direct numerical simulation (DNS) to correctly simulate all the length scale of a fluid dynamics problem as it implies a prohibitive quantity of cells in the mesh. Another technology, large eddy simulation (LES), tries to model the smallest length scales via a theoretical local filtering of the solution in order to reduce the characteristic length scale of the smallest cell needed to correctly simulate the phenomena studied. However, this can still lead to a prohibitive quantity of cells in the mesh. Finally, Reynolds-Averaged-Simulation (RAS), use the RANS equations described in section F to model all the length scales involved in turbulence via a theoretical ensemble averaging. This allows to recover a mean field solution which requires much less cells in the mesh. In this work, we chose to run steady-state RAS to recover mean steady-state fields around airfoils.

Another aspect of the construction of a mesh is the choice of a strategy to resolve the boundary layers close to an obstacle. There are two strategies but each are based on the value of the y^+. This quantity represents a local Reynolds close to the obstacle and is defined as:

$$\begin{cases} y^+ = \frac{yu_\tau}{\nu} \\ u_\tau = \sqrt{\frac{\tau_w}{\rho}} \end{cases} \tag{45}$$

where y is the distance from the wall, τ_w the magnitude of the wall shear stress, ρ the density of the fluid and ν the kinematic viscosity of the fluid. In order to compute the y^+, experimental values on thin plates give the order of magnitude of the wall shear stress term [51]. When we take y as the height of the first cell close to the wall, we can define two strategies:

- *Low-Reynolds simulation:* resolve entirely the boundary layer, $y^+ < 1$
- *High-Reynolds simulation:* model the boundary layer via a so-called wall function, $y^+ \sim 10^2$

As we are interested in accurate force coefficients at the surface of our airfoils, we chose the first strategy to avoid modelling close to the wall. This implies that we need the maximum height y of our first cells close to the wall to be smaller than ν/u_τ. In our case, we chose $y = 2\,\mu\mathrm{m}$ which set the y^+ to be around 1 in the worst case of our design space.

Let us now present the mesh we use for our simulations. This is inspired by the National Aeronautics and Space Administration (NASA) mesh used in [10] to recover experimental force coefficients on the NACA 0012 and 4412. We do not pretend to have the same quality of mesh as the NASA but we still argue that our mesh is well suited for our case. We show it in the next sections by comparing our results to experimental results. Meshes have been generated with the help of *blockMesh*, a hexahedral mesh generator included in the OpenFOAM suite that works by defining blocks. With the help of a dictionnary (namely the *blockMeshDict* file), we set the number of cells and the grading we want to fully determined the meshing inside each block. A scheme of how the domain is divided in multiple blocks and a result of the meshing procedure on the NACA 0012 with an angle of attack of $10°$ is given in Figure 5. More precisely, as in the NASA mesh, a C-Grid domain is defined with a radius and a length of $200\,\mathrm{m}$ (which means 200 chords here). This is smaller than the 500 chords length domain of the NASA but we found it big enough to be insensible to boundary conditions. We now use the index of the nodes, edges and blocks defined on Figure 5. For the edges 310, 411, 58, 69 and 710, the smallest cell is of height $2\,\mu\mathrm{m}$ as already said above and we set the expansion ratio (the length ratio between two consecutive cells) to 1.075. For the edges 01 and 12 the smallest cell is of height $100\,\mu\mathrm{m}$ and the expansion ratio is also set to 1.075. At the upper surface of the airfoil, at the leading edge, the smallest cell is of width $10\,\mu\mathrm{m}$ (at node 8) and we set the expansion ratio to 1.025 until roughly the maximum of camber of the airfoil (at node 11). From node 11 to node 10, an automatic expansion ratio is computed to fill the entire segment. Almost the same procedure is applied at the lower surface of the airfoil, the only difference is that, for consistency, the expansion ratio between node 9 and 10 is set such that the last cells (at node 10) is of the same width as the one at the upper surface. Edge 34 or 67 have a fix grading of 1 and edge 45 or 56 have a grading such that the width of the cell at the junction of blocks 2 and 3 or 4 and 5 are the same (this is only true at node 4 or 6 and a significant width difference can be seen at the center of edges 411 or 69). Finally edges

25
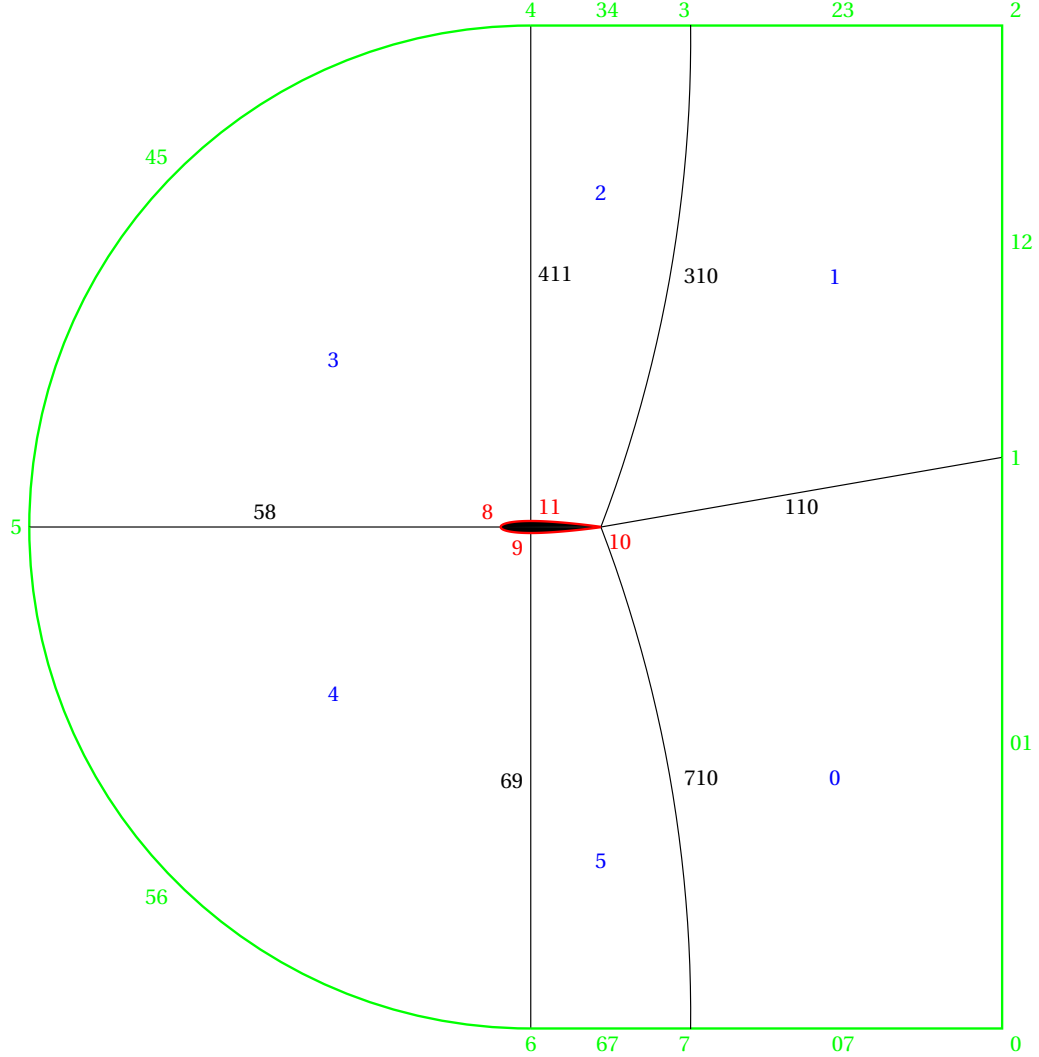
Figure 5: Scheme of the mesh template. This is a scheme for the NACA 0012 with an angle of attack of $10°$. The aerofoil patch is highlighted in red, the freestream patch is highlighted in green and the internal patch is the union of the blocks 0 to 5 highlighted in blue. The indices of nodes and blocks are the same as in the *blockMeshDict* file.

07, 110 and 23 have a smallest cell of the same width as for block 2 or 5 (with the same remark, this is true only at nodes 3, 7 and 10) and a grading of 1.075 is applied.

In Figure 6 we give the number of cells and nodes in simulations of the dataset. We also give those quantities for the cropped simulations used for the ML tasks.

J Boundary Conditions

In this section, we explicit the different boundary conditions set on the different patches of the mesh. The quantities needed for a simulation depend on whether we run incompressible or compressible physics and on the turbulence model chosen. In Table 7, 8 and 9 we give the different OpenFOAM settings used for the different fields involved in the simulation, that are:

- U : ensemble averaged velocity in $\mathrm{m\,s^{-1}}$
- p : ensemble averaged effective pressure in Pa (in the incompressible case the pressure is divided by the density ρ, $p \rightarrow p/\rho$)

26
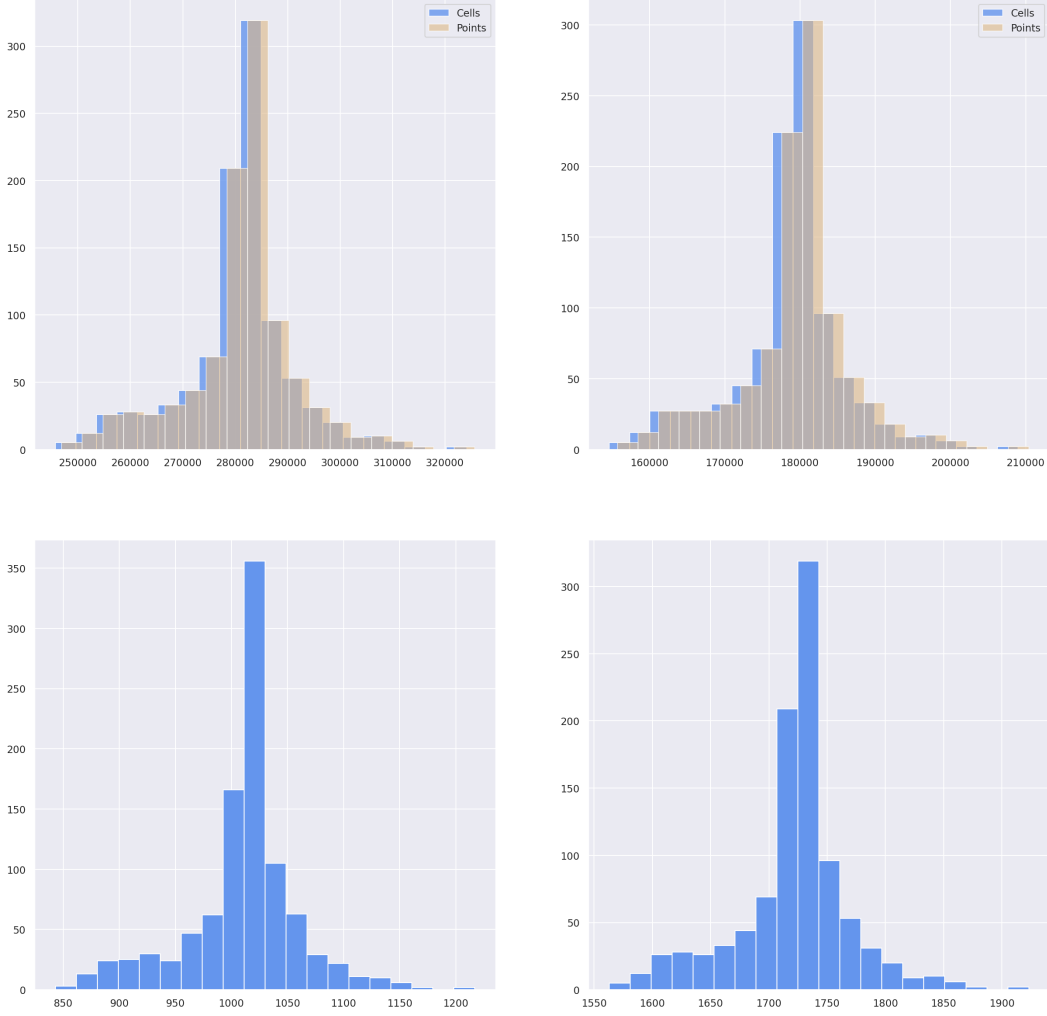
Figure 6: Histograms of the number of cells and nodes in simulations of the dataset. (top left) Number of cells and nodes in internal meshes for CFD simulations (top right) Numbers of cells and nodes in internal meshes for cropped simulations (bottom left) Number of nodes on airfoils patches (bottom right) Number of nodes on freestream patches. For the bottom plots, we only give the number of nodes as they are equal to the number of cells.

- ν_t : kinematic turbulent viscosity in $\mathrm{m^2\,s^{-1}}$
- $\tilde{\nu}^{\parallel}$: Spalart-Allmaras variable in $\mathrm{m^2\,s^{-1}}$
- $k^{\dagger}$: turbulent kinetic energy in J
- $\omega^{\dagger}$: specific dissipation rate via turbulence in $\mathrm{s^{-1}}$
- $T^{\ddagger}$: temperature in K
- $\alpha_t{}^{\ddagger}$: turbulent thermal diffusivity in $\mathrm{m^2\,s^{-1}}$

We do not present here the discretization schemes chosen for the simulations, nor the linear solver and hyperparameters of the SIMPLE algorithm. You can find them in the *fvSchemes* and *fvSolution* dictionnaries respectively. We just mention that we used the SIMPLEC [16] algorithm for the incompressible case and the classical SIMPLE [8] one for the compressible setup as the SIMPLEC was not stable in this case.

[∥]Only for the Spalart-Allmaras turbulent model.

[†]Only for the $k - \omega$ SST turbulent model [42].

[‡]Only in the case of compressible simulations.

27

Table 7: Boundary conditions set on the different patches of the mesh for compressible and incompressible simulations. Values of the constants are given for the air at sea level and at 298.15 K.

Fields	Internal	Aerofoil	Freestream
U	U_∞	noSlip	freestreamVelocity
p	$0^\star$	zeroGradient	freestreamPressure
ν_t	ν	nutLowReWallFunction	freestream
$\tilde{\nu}$	4ν	fixedValue	freestream
k	$0.001U_\infty^2/Re_L$	fixedValue	freesteam
ω	$5U_\infty/L$	omegaWallFunction	freestream
T	$298.15\,\mathrm{K}$	zeroGradient	freestream
α_t	ν_t/Pr_t	compressible::alphatWallFunction	calculated

* This value has to be set to an absolute pressure value, in our case $1.013 \times 10^5\,\mathrm{Pa}$, for the compressible case.

Table 8: Definition of the quantities involved in Table 7 and their values for the air at sea level and at a temperature of 298.15 K (25 °C).

Quantity	Definition	Value
ρ	Density of the fluid	$1.184\,\mathrm{kg\,m^{-3}}$
ν	Kinematic viscosity of the fluid	$1.56 \times 10^{-5}\,\mathrm{m^2\,s^{-1}}$
L	Length of the domain	$400\,\mathrm{m}$
U_∞	Velocity at the inlet	-
Re_L	Reynolds number computed with L	$U_\infty L/\nu$
Pr_t	Turbulent Prandtl number (constant)	0.85

K   Simulation Validation

In this section, we test our mesh and boundary conditions on two different problems, with two turbulence models and in the compressible and incompressible settings, in order to validate the choice made in this work. To do so, we use the experimental data produced by the NASA and available on the Turbulence Modeling Resource (TMR) website of the Langley Research Center [10] for the NACA 0012 and 4412.

NACA 0012 airfoil.   We compare our results with experimental data for the force coefficients done on the NACA 0012 [12, 33]. We restricted our study to the case of a Reynolds of 6 million for different angle of attacks (see Table XIII [33]). In our simulations, we run an incompressible solver with the properties of the air at 298.15 K and at sea level (see Table 8) which gives an inlet velocity U_∞ of $93.6\,\mathrm{m\,s^{-1}}$ with a characteristic length equal to the chord of the airfoil (in our case $1\,\mathrm{m}$). The celerity of sound in this medium is taken to be $346.1\,\mathrm{m\,s^{-1}}$, which gives a Mach number ($Ma := U_\infty/c$) of roughly 0.27. We often set a limit a 0.3 for the Mach number in order to run incompressible simulations, and as we are close to this limit, we run incompressible and compressible simulations for an additional restricted set of angle of attacks of $0°$ and $10°$. The pressure coefficient at the surface of the airfoil are compared to another set of experimental data (Table II of [12]) done at Mach 0.3 and Reynolds 6 million for this two angles of attack (more precisely at angle $0.0169°$ and $10.0254°$). Moreover, we tried with the Spalart-Allmaras and $k - \omega$ SST models of turbulence.

The pressure coefficient c_p at the surface of the airfoil is a dimensionless coefficient defined as:

$$c_p := \frac{\bar{p} - \bar{p}_\infty}{q_\infty}, \qquad q_\infty := \frac{1}{2}U_\infty^2 A \tag{46}$$

where $\bar{p}$ is the mean-field reduced pressure, $\bar{p}_\infty$ the far field pressure (set to 0 in the incompressible case), U_∞ is the magnitude of the inlet velocity and A is the characteristic area of the problem, we take here $A = 1\,\mathrm{m^2}$.

28

Table 9: Definition of the boundary conditions involved in Table 7 and values we use when asked by OpenFOAM. The values given for *fixedValue* are the values of k and $\tilde{\nu}$ at the surface of the airfoil. The quantity $\beta_1 = 0.075$ is a constant of the $k - \omega$ model [66] and $\Delta y = 2\,\mu m$ the height of the first cells of the boundary layer.

Boundary condition	Definition	Value
fixedValue	Set the quantity to a constant	$0\,\mathrm{J}/0\,\mathrm{m^2\,s^{-1}}$
calculated	Derived from other quantities	Internal field value
noSlip	Set the velocity to 0	-
zeroGradient	Set the field at the boundary to the value of the internal field	-
freestream	Mixed boundary condition between *fixedValue* and *zeroGradient* depending on the direction of the flux	Internal field value
freestreamVelocity	Same as *freestream* but switches in accordance with *freestreamPressure*	Internal field value
freestreamPressure	Same as *freestream* but switches in accordance with *freestreamVelocity*	Internal field value
nutLowReWallFunction	Set the turbulent viscosity to 0	$0\,\mathrm{m^2\,s^{-1}}$
omegaWallFunction	For low Reynolds simulation, equivalent to *fixedValue*	$\frac{6\nu}{\beta_1 \Delta y^2}$
compressible::alphatWallFunction	Equivalent to *fixedValue* with a value of ν_t/Pr_t	$0\,\mathrm{m^2\,s^{-1}}$

In Figure 7 and 8, the surface pressure coefficient is given and we see no significant difference between the two models nor between the compressible and incompressible cases. All the simulations are in good agreement with the experiments.

In Figure 9 are displayed the drag and lift coefficients with respect to angle of attacks and the drag coefficient with respect to the lift coefficient for the two models and the experiments. In the compressible case, only $0°$ and $10°$ have been simulated for time and stability reasons, no significant differences are present with the incompressible simulations. In the incompressible case, a missing point in the plot means that the simulation was unstable and we did not manage to make it converge correctly. We can see that both compressible and incompressible solver gives a slightly over estimated drag with respect to experiments, this is in agreement with the TMR. We also see that the $k - \omega$ SST model is more stable than the Spalart-Allmaras model, we noticed a faster convergence for the first too. Finally, the $k - \omega$ SST model fits better the experiments than the Spalart-Allamaras model. In total, both model are in good agreement with experimental data but the $k - \omega$ SST model looks more stable, faster to converge and more accurate than the Spalart-Allamaras.

From this point, we only run incompressible simulations as this validation case showed no distinctions between compressible and incompressible simulations. We now test our setup on another validation case in order to chose between the two turbulence models.

NACA 4412 airfoil. In this setup, the experimental data [61] are done with a NACA 4412 at an angle of attack of $13.87°$ and a Reynolds number of 1.52 million. The values of the experimental data are the one given on the NACA 4412 page of the TMR. The values found on this website are slightly different from the one found in the original papers, moreover, the normalization factor for the pressure coefficient is computed with a reference velocity U_{ref} of roughly $0.93U_\infty$ but, as in

29
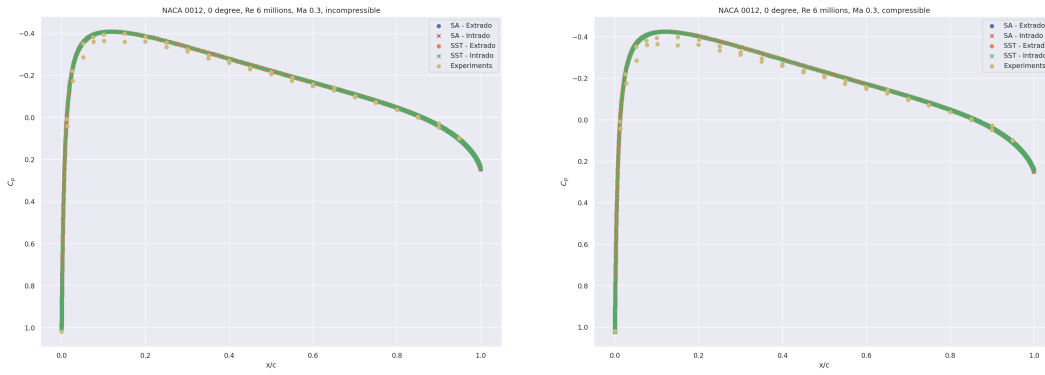
Figure 7: Pressure coefficient at the surface of the airfoil for the NACA 0012 at an angle of attack of $0°$ in the incompressible (left) and compressible (right) cases for the Spalart-Allamaras, $k - \omega$ SST models and the experiments with respect to the abscissas in chord length. The points on the upper and lower surfaces are given in different colors for the simulations.
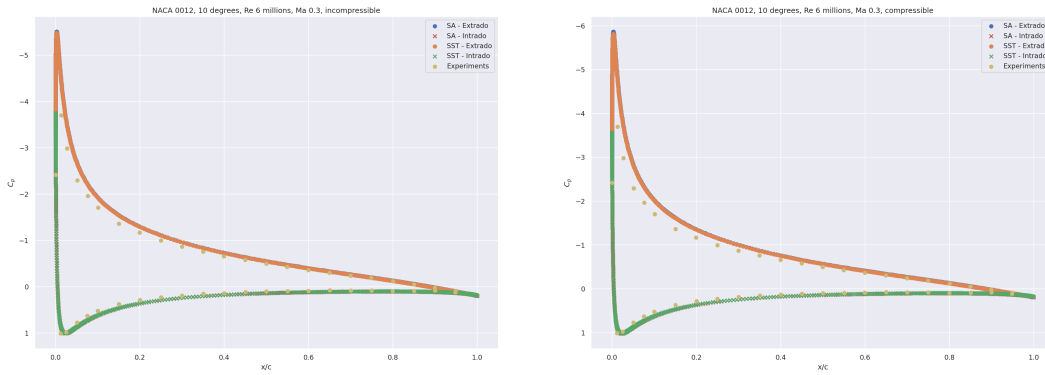
Figure 8: Pressure coefficient at the surface of the airfoil for the NACA 0012 at an angle of attack of $10°$ in the incompressible (left) and compressible (right) cases for the Spalart-Allamaras, $k - \omega$ SST models and the experiments with respect to the abscissas in chord length. The points on the upper and lower surfaces are given in different colors for the simulations.
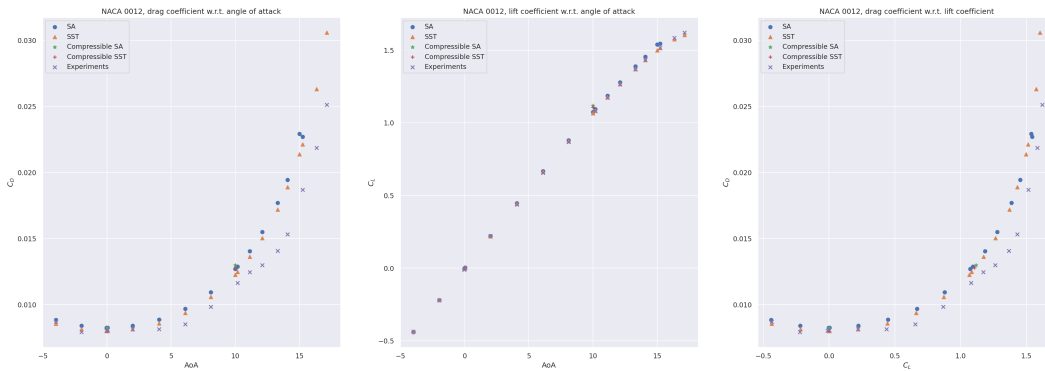
Figure 9: Drag and lift coefficients with respect to angle of attacks and to each other in the case of the NACA 0012. The compressible simulations have only be done at $0°$ and $10°$.

30
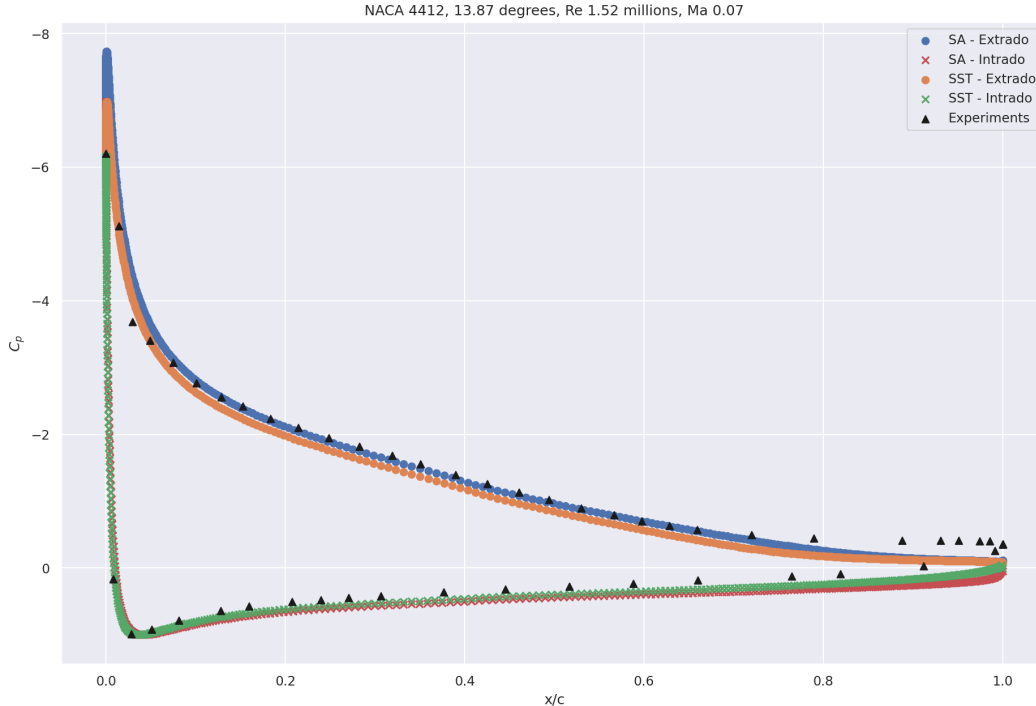
Figure 10: Pressure coefficient at the surface of a NACA 4412 with an angle of attack of $13.87°$ and a Reynolds of 1.52 million. The normalization of the pressure coefficient is computed with the magnitude of the inlet velocity U_∞. Are displayed the experimental data, the Spalart-Allmaras and $k - \omega$ SST models incompressible results.

the NASA simulations, the results better fit when using a normalization factor computed with U_∞. This is underlined on the TMR page and incite us to take this validation case only as a qualitative validation.

In Figure 10, the pressure coefficient is given with a normalization factor computed with the magnitude of the inlet velocity U_∞. Both turbulence models results are in good agreement with the experiments.

In Figure 11, we look at the boundary layer of the airfoil at different abscissas. Here the x and y components of the velocity (denoted by u and v respectively) are normalized by U_{ref} and the term $u'v'$, corresponding to the shear stress term of the Reynolds stress tensor, is normalized by U_{ref}^2. We start each plot at a given point at the surface of the airfoil and take the direction of the normal of the airfoil at this point. Hence, the name $(y - y_0)/c$ for the ordinate of the plot has to be understand as the distance to the airfoil in the normal direction in chord length. Both turbulence models have difficulties to predict correctly the experimental data, this behaviour has already been pointed out in the TMR study of the NACA 4412 and our results are in good agreement with theirs. Moreover, the $k - \omega$ SST model seems to give more realistic results than the Spalart-Allmaras one.

In total, our simulations on the NACA 0012 and 4412 are in good (at least qualitatively) agreement with the experiments. The incompressible $k - \omega$ SST model setup seems the best candidate for fast, stable and high fidelity simulations. In this work, we keep this setup to generate the dataset.

L   Models architecture

For all of the tasks, the same architecture is used in addition with the same hyperparameters. Each model is preceded by an encoder and followed by a decoder both defined as MLP with ReLU activation function, no batch normalization, and with $7 - 64 - 64 - 8$ and $8 - 64 - 64 - 4$ neurons respectively, meaning an dimension of encoding of 8. Those encoder and decoder are trained together with the chosen model.

31
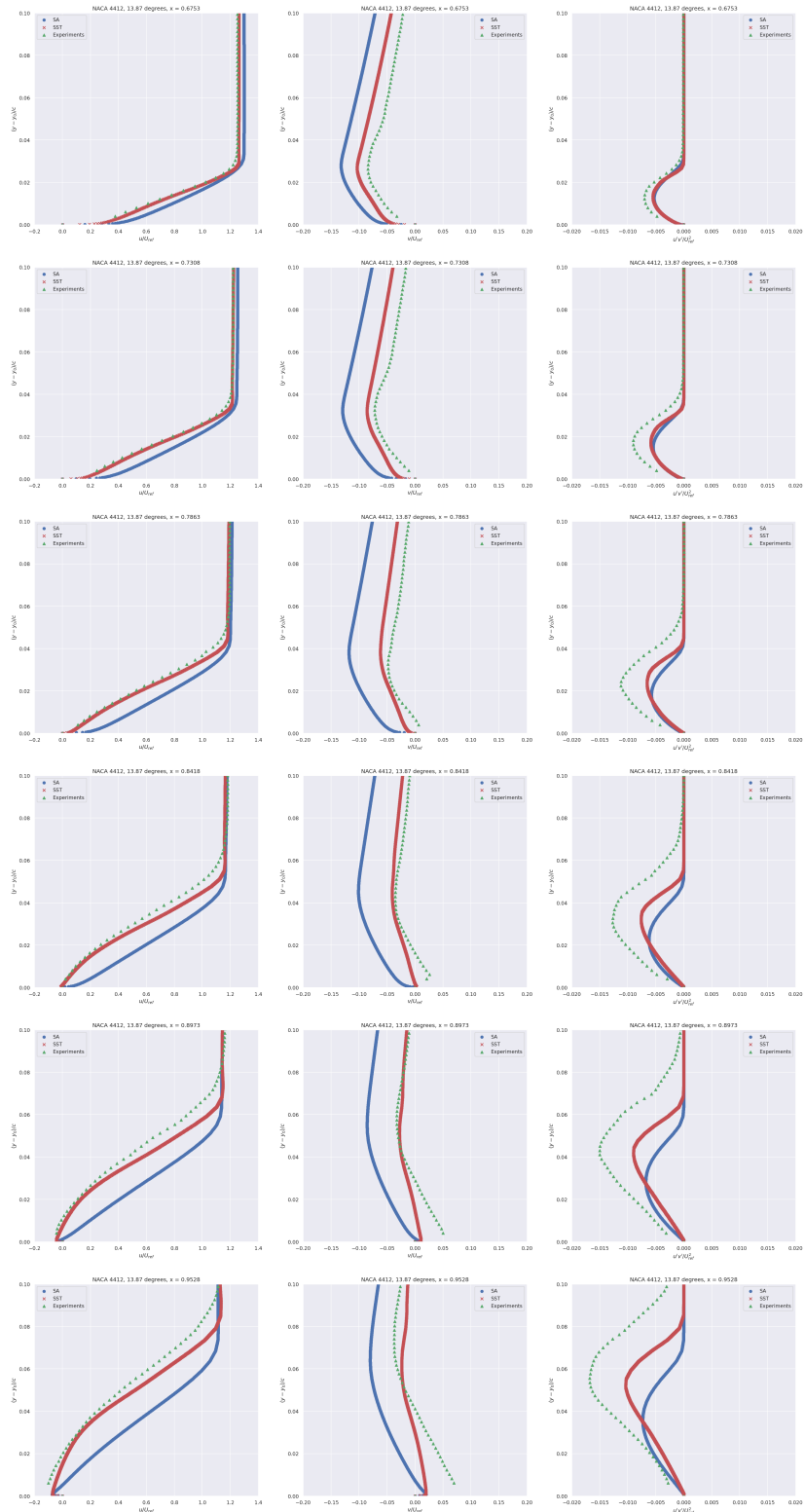
Figure 11: Boundary layer velocity components and shear Reynolds stress for different point at the surface of the NACA 4412 at a Reynolds number of 1.52 million. Each quantity is normalized either by U_{ref} or U_{ref}^2. The ordinate has to be understand as the distance to the given point at the surface of the airfoil following the normal direction.

32

Multi-Layer Perceptron. The first baseline is another MLP with ReLU activation function and batch normalization before the activation. It has $8 - 64 - 64 - 64 - 8$ neurons.

GraphSAGE. The GraphSAGE acts on a radius graph of 32000 nodes and radii $5\,\mathrm{cm}$. It is defined with 3 hidden layers and 64 hidden features per node.

PointNet. The PointNet is copied from the segmentation task of [11]. We chose 8 neurons as a base number and we did not include any batch normalization nor dropout as it was performing badly with.

Graph U-Net. For the Graph U-Net, we defined it with five scales, downsampling by half at each scale and multiplying by two the number of features at each scales. The radii of the radius graphs are $5\,\mathrm{cm}$, $20\,\mathrm{cm}$, $50\,\mathrm{cm}$, $1\,\mathrm{m}$ and $10\,\mathrm{m}$. The last radii is chosen such that the graph at the coarsest scale is fully connected. Each of those radius graphs have a limit of 64 neighbors per node. For the downsampling, we did not use the gPool method presented in the historical paper [23] and replaced it by a random downsampling over the remaining nodes, recreating a radius graph afterwards. This leads to better results. On the upward pass, we chose to aggregate the different informations from the skip connection and the preceding scale by concatenating the features. Finally, we chose to start with 8 features at the finest scale.

The learning rate for all of those experiments is set with a one-cycle cosine [55] rate of maximum 0.001, simulations are fed one by one to the different models during training (*i.e.* 32000 nodes with an associated radius graph when needed) and the number of epochs is chosen such that for each task, we have the same number of gradient updates:

- *Full data regime:* 400 epochs
- *Scarce data regime:* 1600 epochs
- *Reynolds extrapolation regime:* 635 epochs
- *Angle of attack extrapolation regime:* 398 epochs

Ultimately, the different models are trained on 90% of the predefined training set of those different regime, the last 10% have been used as a validation set.

M Additional Results

In this section, we treat the results of the three remaining machine learning tasks. Those tasks are not less important than the full data regime treated in the main paper, they are actually more important than the latter. Those regime better represent the challenges of real-life problems as the data is often lacking and extrapolation is often sought. We first give the missing results in the full data regime and then present the three other regimes.

Full data regime. In Figure 13 we show the pressure and skin friction coefficients distributions over the surface of five randomly chose airfoils in the test set. In Figure 12, we present the x and y components of the velocity distribution in the boundary layer at the upper surface of the same five airfoils. Those velocity profiles are given at four abscissas: $x = 0.2\,\mathrm{m}$, $x = 0.4\,\mathrm{m}$, $x = 0.6\,\mathrm{m}$ and $x = 0.8\,\mathrm{m}$.

As we said in Section 5, the wall shear stress is largely overestimated, this behaviour is understood via the boundary layer velocity profiles where the first inferred point after the surface is badly approximated. This leads to high errors on the drag coefficient.

Scarce data regime. In this regime, we test on the same test set as the full data regime but we trained with only two hundred simulations instead of eight hundreds.

In Table 10, we give the MSE over the volume and at the surface of airfoils for the different regressed fields. In Table 11 we give the mean relative errors on the force coefficient and the Spearman's rank correlation coefficient. In Figure 14 we plot the predicted force coefficients with respect to the true coefficients. In Figure 15, we plot the velocity and turbulent viscosity profiles in the boundary layer for randomly chosen test geometries and in Figure 16 the surface coefficients for the same geometries.

33
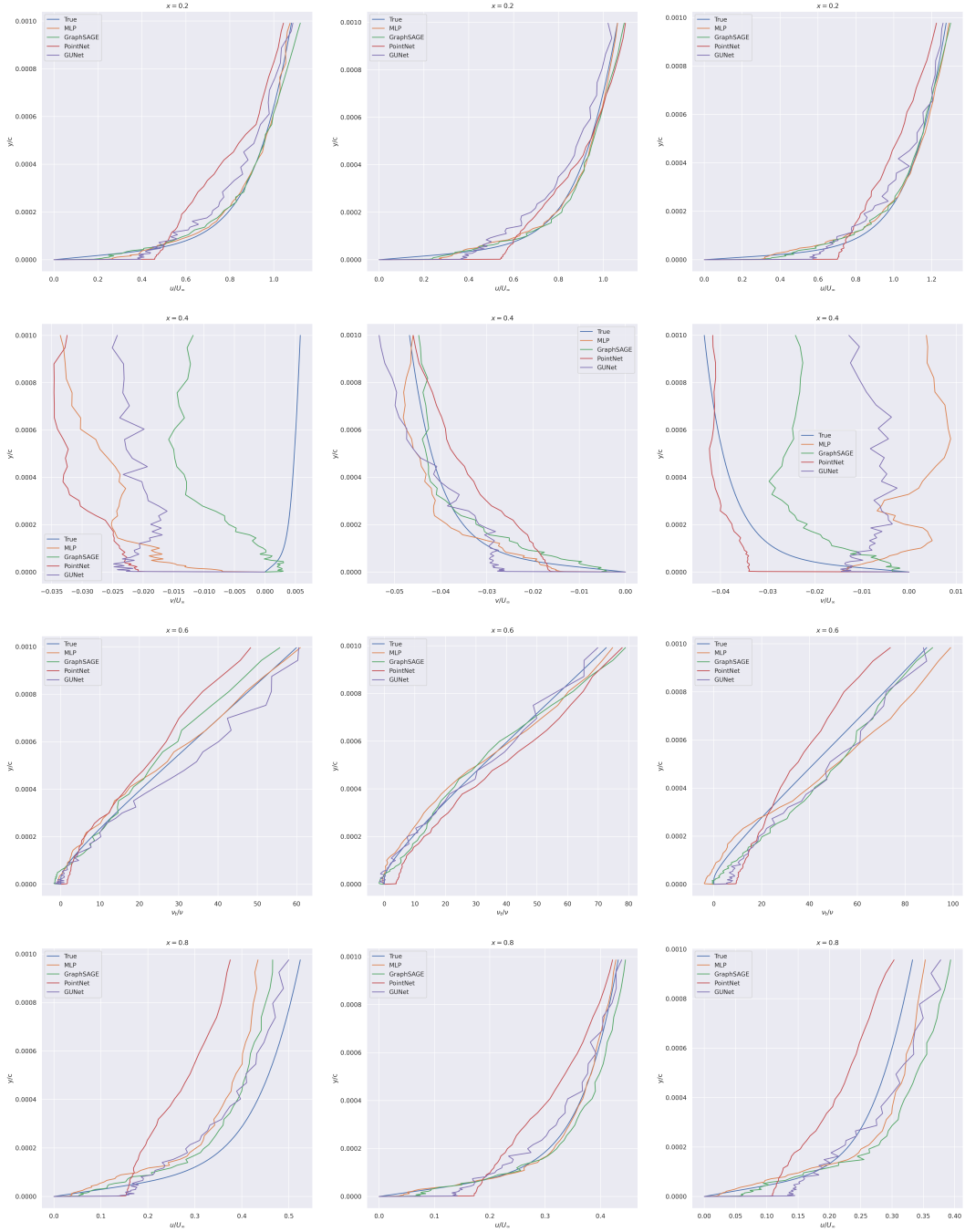
Figure 12: Comparison of the predicted boundary layers profiles on three random test geometries at different abscissas in the full data regime with respect to the true ones. Each column of plots represent a different airfoil and each line of plots represent a different abscissas. The x and y component of the velocity are denoted by u and v respectively and the turbulent viscosity is denoted by ν_t. Each quantity is normalized either by u_∞ the inlet velocity magnitude or ν the fluid viscosity.

34
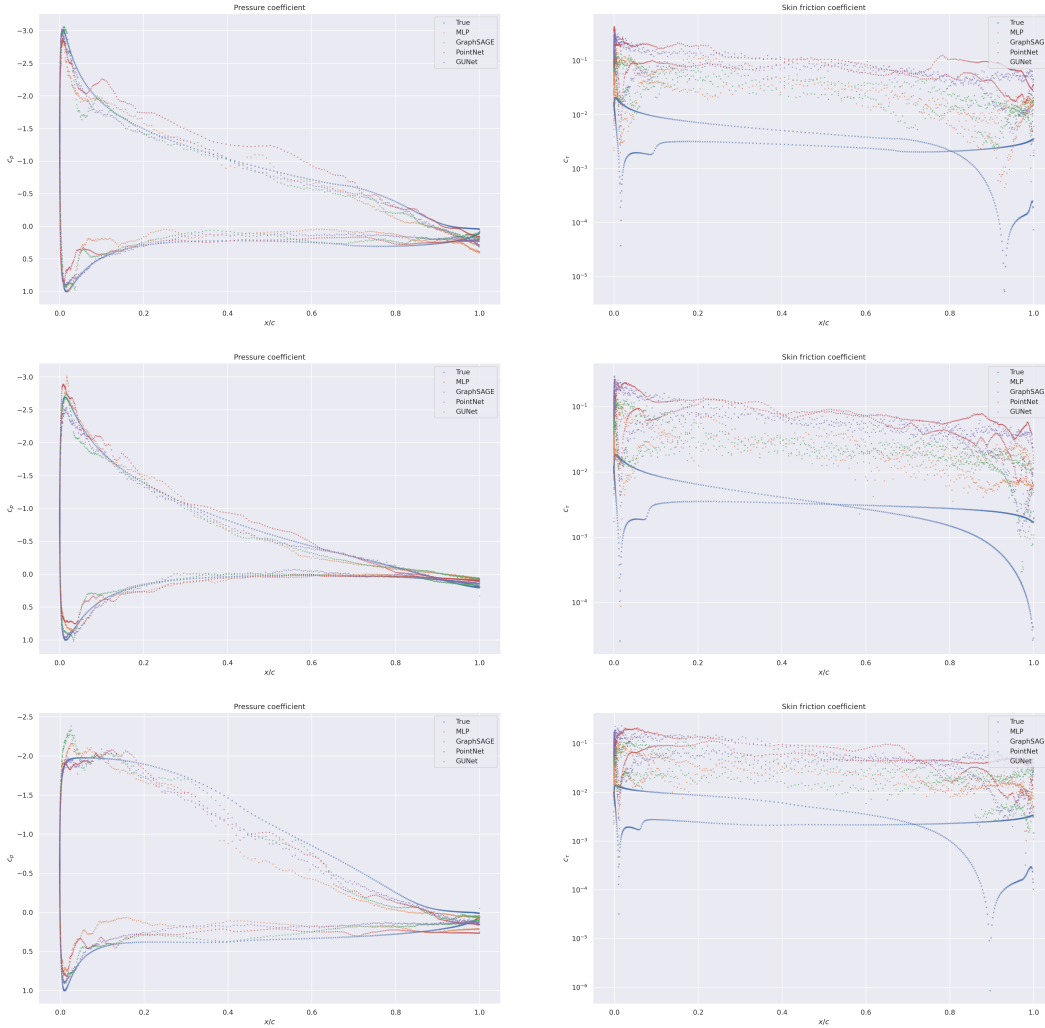
Figure 13: Comparison of the predicted surface coefficients profiles on three random test geometries in the full data regime with respect to the true one. (left) Surface coefficient c_p (right) Skin friction coefficient c_τ. Each line of plots represents a different airfoil. Skin friction coefficient plots are given in log scale.

Table 10: Mean squared error on the different normalized fields for an MLP and standard GDL baselines on the test set in the scarce data regime. Only the reduced pressure is given on the surface as the other quantities are null via the boundary conditions. Those quantities are directly regressed by the models.

Model	Volume				Surface
	$\bar{u}_x$ ($\times 10^{-2}$)	$\bar{u}_y$ ($\times 10^{-2}$)	$\bar{p}$ ($\times 10^{-2}$)	ν_t ($\times 10^{-2}$)	$\bar{p}$ ($\times 10^{-1}$)
MLP	1.65±0.03	**1.45±0.07**	3.90±0.57	5.01±0.76	2.19±0.53
GraphSAGE	**1.46±0.13**	1.45±0.12	4.70±0.80	6.11±0.79	1.95±0.34
PointNet	3.11±0.30	2.78±0.39	**3.29±1.05**	5.58±2.36	1.83±0.41
Graph U-Net	1.75±0.19	1.83±0.18	3.39±0.84	**4.30±1.00**	**1.47±0.35**

35

Table 11: Relative errors (Spearman's rank correlation) for the predicted drag coefficient C_D (ρ_D) and lift coefficient C_L (ρ_L) in the scarce data regime. We want the Spearman's correlation to be close to one. Those quantities are computed as a post processing from the unnormalized regressed fields.

Model	Relative error		Spearman's correlation	
	C_D	C_L	ρ_D	ρ_L
MLP	**2.95±0.14**	0.66±0.16	-0.24±0.08	0.923±0.026
GraphSAGE	3.50±1.00	**0.39±0.10**	-0.14±0.18	**0.981±0.006**
PointNet	8.35±1.39	0.59±0.13	-0.05±0.27	0.949±0.019
Graph U-Net	6.87±1.80	0.42±0.13	-0.10±0.23	0.976±0.009
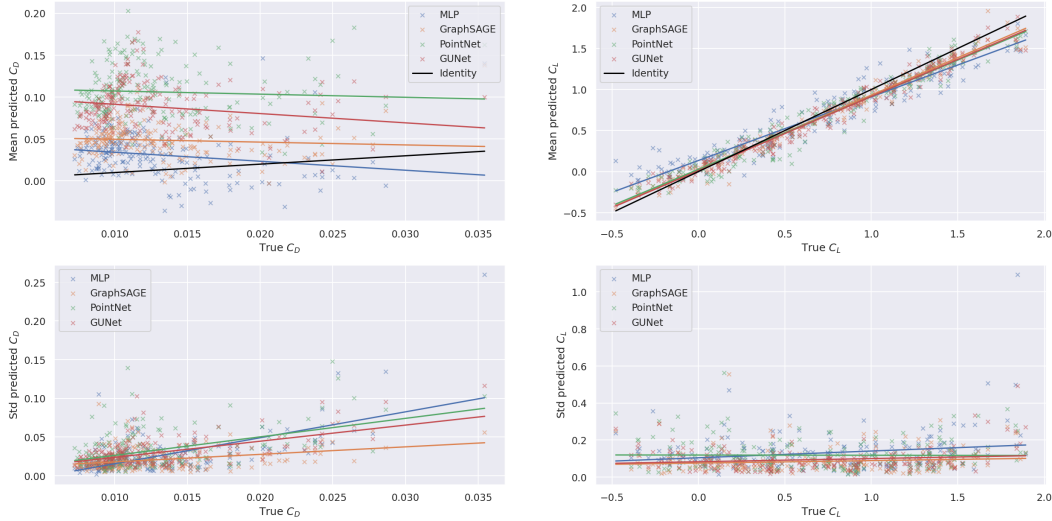
Figure 14: Predicted drag (left) and lift (right) coefficients with respect to the true ones in the scarce data regime. The mean (top) and standard deviation (bottom) of each point on the five copy of the trained models are separated for sake of readability. A linear regression is done for each point cloud in order to highlight linear trends. On the top plots, the Identity graph is given in black for comparison.

Reynolds extrapolation regime. In this regime, we test on out of distribution Reynolds number.

In Table 12, we give the MSE over the volume and at the surface of airfoils for the different regressed fields. In Table 13 we give the mean relative errors on the force coefficient and the Spearman's rank correlation coefficient. In Figure 17 we plot the predicted force coefficients with respect to the true coefficients. In Figure 18, we plot the velocity and turbulent viscosity profiles in the boundary layer for randomly chosen test geometries and in Figure 19 the surface coefficients for the same geometries.

Table 12: Mean squared error on the different normalized fields for an MLP and standard GDL baselines on the test set in the Reynolds extrapolation regime. Only the reduced pressure is given on the surface as the other quantities are null via the boundary conditions. Those quantities are directly regressed by the models.

Model	Volume				Surface
	$\bar{u}_x$ ($\times 10^{-2}$)	$\bar{u}_y$ ($\times 10^{-2}$)	$\bar{p}$ ($\times 10^{-2}$)	ν_t ($\times 10^{-1}$)	$\bar{p}$ ($\times 10^{-1}$)
MLP	9.51±1.27	4.92±0.80	4.30±0.19	1.31±0.34	20.9±35.5
GraphSAGE	**7.56±1.05**	**3.50±0.61**	**3.83±0.25**	1.69±0.38	**1.80±0.34**
PointNet	9.42±1.08	7.13±0.80	4.01±0.74	**1.27±0.44**	2.01±0.76
Graph U-Net	8.38±1.82	5.25±1.36	4.48±0.40	1.28±0.31	2.06±0.44

36
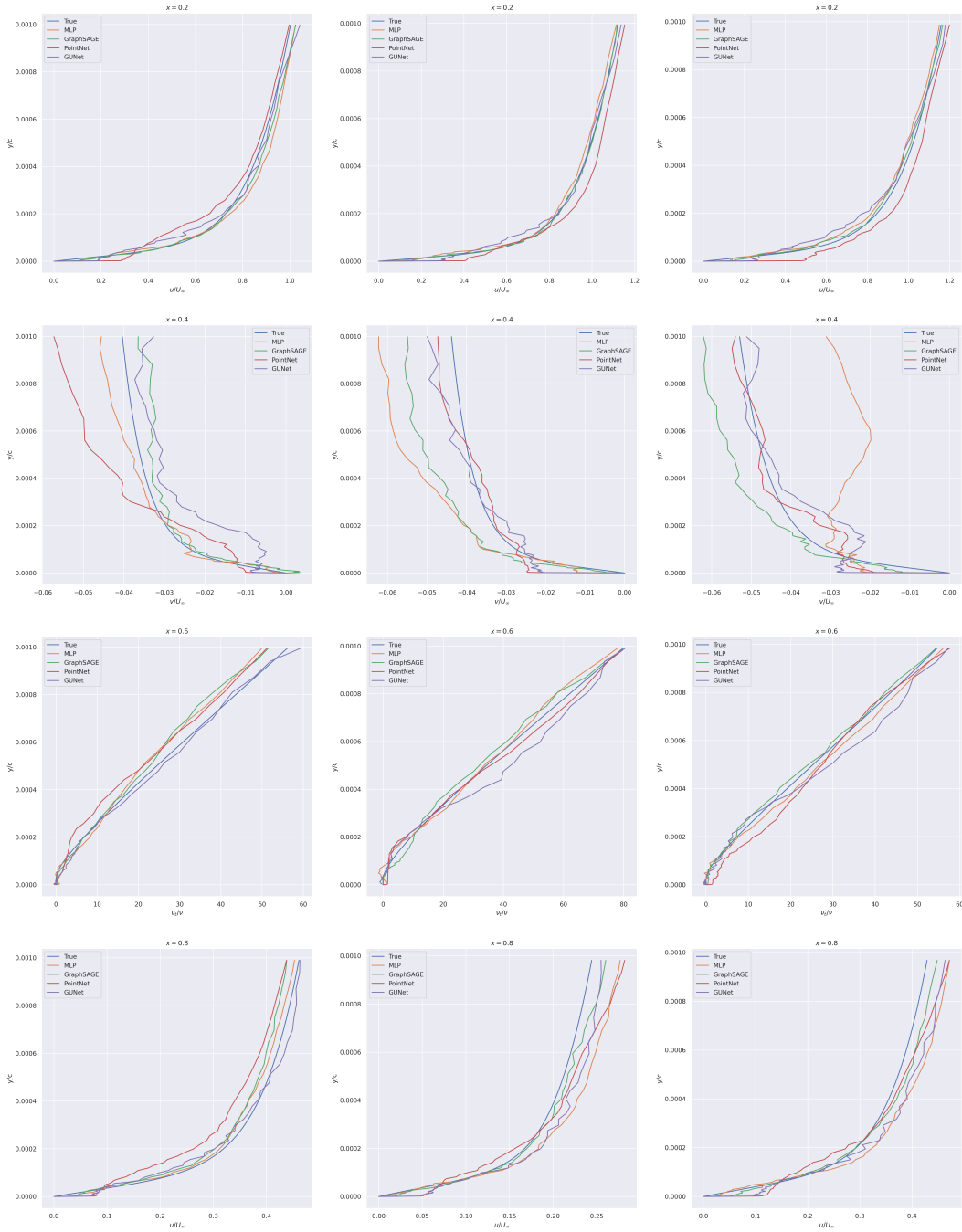
Figure 15: Comparison of the predicted boundary layers profiles on three random test geometries at different abscissas in the scarce data regime with respect to the true ones. Each column of plots represent a different airfoil and each line of plots represent a different abscissas. The x and y component of the velocity are denoted by u and v respectively and the turbulent viscosity is denoted by ν_t. Each quantity is normalized either by u_∞ the inlet velocity magnitude or ν the fluid viscosity.

37
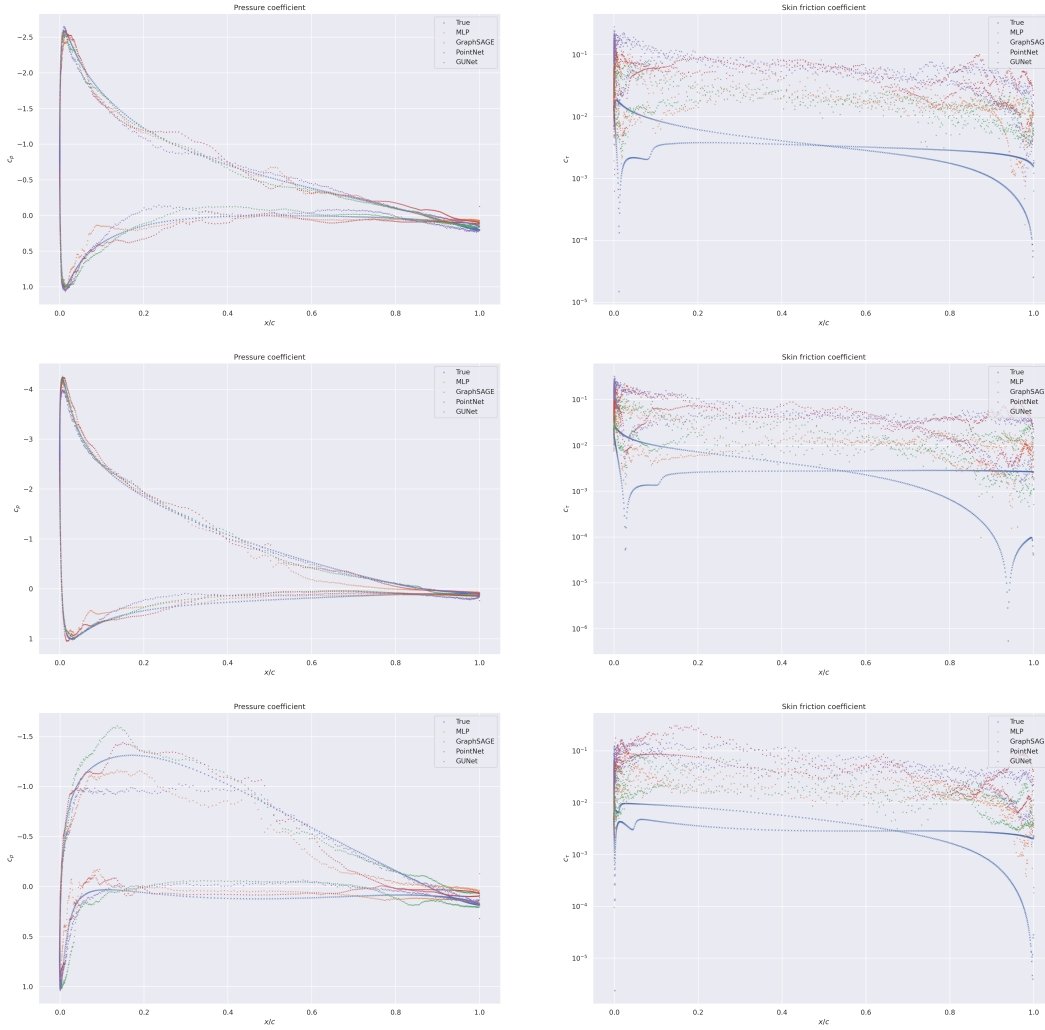
Figure 16: Comparison of the predicted surface coefficients profiles on three random test geometries in the scarce data regime with respect to the true one. (left) Surface coefficient c_p (right) Skin friction coefficient c_τ. Each line of plots represents a different airfoil. Skin friction coefficient plots are given in log scale.

Table 13: Relative errors (Spearman's rank correlation) for the predicted drag coefficient C_D (ρ_D) and lift coefficient C_L (ρ_L) in the Reynolds extrapolation regime. We want the Spearman's correlation to be close to one. Those quantities are computed as a post processing from the unnormalized regressed fields.

Model	Relative error		Spearman's correlation	
	C_D	C_L	ρ_D	ρ_L
MLP	13.4±7.2	3.33±3.29	-0.15±0.15	0.642±0.274
GraphSAGE	**8.97±1.28**	**0.62±0.12**	0.01±0.06	**0.927±0.027**
PointNet	11.6±2.8	0.90±0.33	0.01±0.24	0.898±0.056
Graph U-Net	13.3±1.8	0.87±0.37	0.03±0.12	0.904±0.064

38
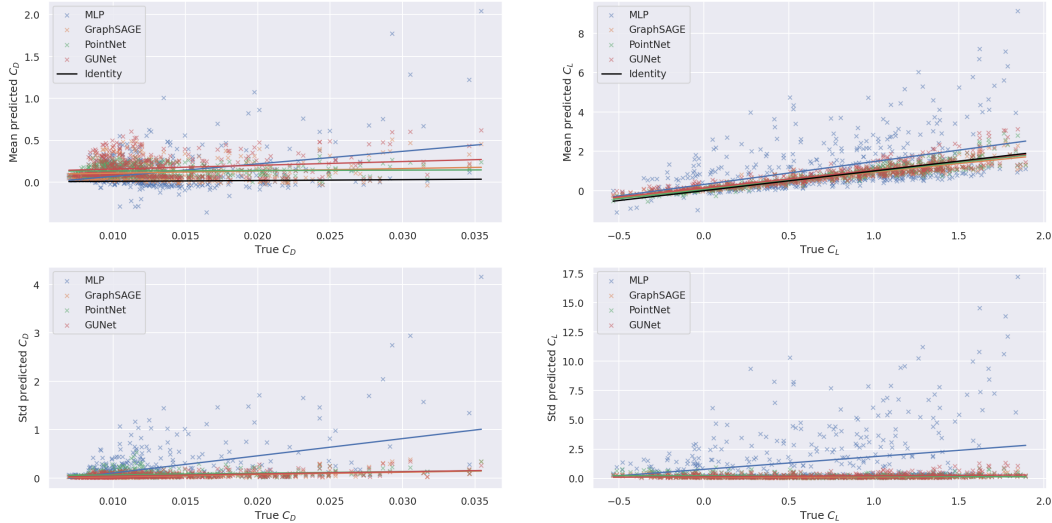
Figure 17: Predicted drag (left) and lift (right) coefficients with respect to the true ones in the Reynolds extrapolation regime. The mean (top) and standard deviation (bottom) of each point on the five copy of the trained models are separated for sake of readability. A linear regression is done for each point cloud in order to highlight linear trends. On the top plots, the Identity graph is given in black for comparison.

Table 14: Mean squared error on the different normalized fields for an MLP and standard GDL baselines on the test set in the angle of attack extrapolation regime. Only the reduced pressure is given on the surface as the other quantities are null via the boundary conditions. Those quantities are directly regressed by the models.

Model	Volume				Surface
	$\bar{u}_x\,(\times 10^{-2})$	$\bar{u}_y\,(\times 10^{-1})$	$\bar{p}\,(\times 10^{-1})$	$\nu_t\,(\times 10^{-1})$	$\bar{p}\,(\times 10^{-1})$
MLP	6.96±0.54	1.06±0.17	1.17±0.25	5.43±0.24	8.76±1.59
GraphSAGE	**4.43±0.33**	**0.94±0.22**	**1.09±0.22**	5.18±0.37	7.64±0.95
PointNet	8.68±2.34	1.58±0.54	1.62±0.66	**4.63±0.40**	**5.85±0.36**
Graph U-Net	5.69±0.71	1.03±0.28	1.49±0.75	5.35±0.59	6.97±2.32

Angle of attack extrapolation regime. In this regime, we test on out of distribution angle of attacks.

In Table 14, we give the MSE over the volume and at the surface of airfoils for the different regressed fields. In Table 15 we give the mean relative errors on the force coefficient and the Spearman's rank correlation coefficient. In Figure 20 we plot the predicted force coefficients with respect to the true coefficients. In Figure 21, we plot the velocity and turbulent viscosity profiles in the boundary layer for randomly chosen test geometries and in Figure 22 the surface coefficients for the same geometries.

Summary. All the three latter regimes are obviously more difficult than the full data one. The MSE scores on the scarce data regime, the Reynolds and angle of attack extrapolation regimes are order of magnitude greater than the ones on the full data regime. This can lead to higher relative errors on force coefficients and lower score on Spearman's correlation but is not directly linked to (as in the scarce data regime). In any case, the wall shear stress is never well predict as the velocity profiles in the boundary layer are not accurate close to the geometries and the pressure coefficient suffers from out of distribution conditions depending on the models. In total, there is a lot of progress possible with those four settings and we hope that more recent architectures and technologies will better perform on the proposed metrics.

In Table 16 we give the MSE scores of all of the models on all of the tasks and in Table 17 their scores on the force coefficients in a more readable way.

39
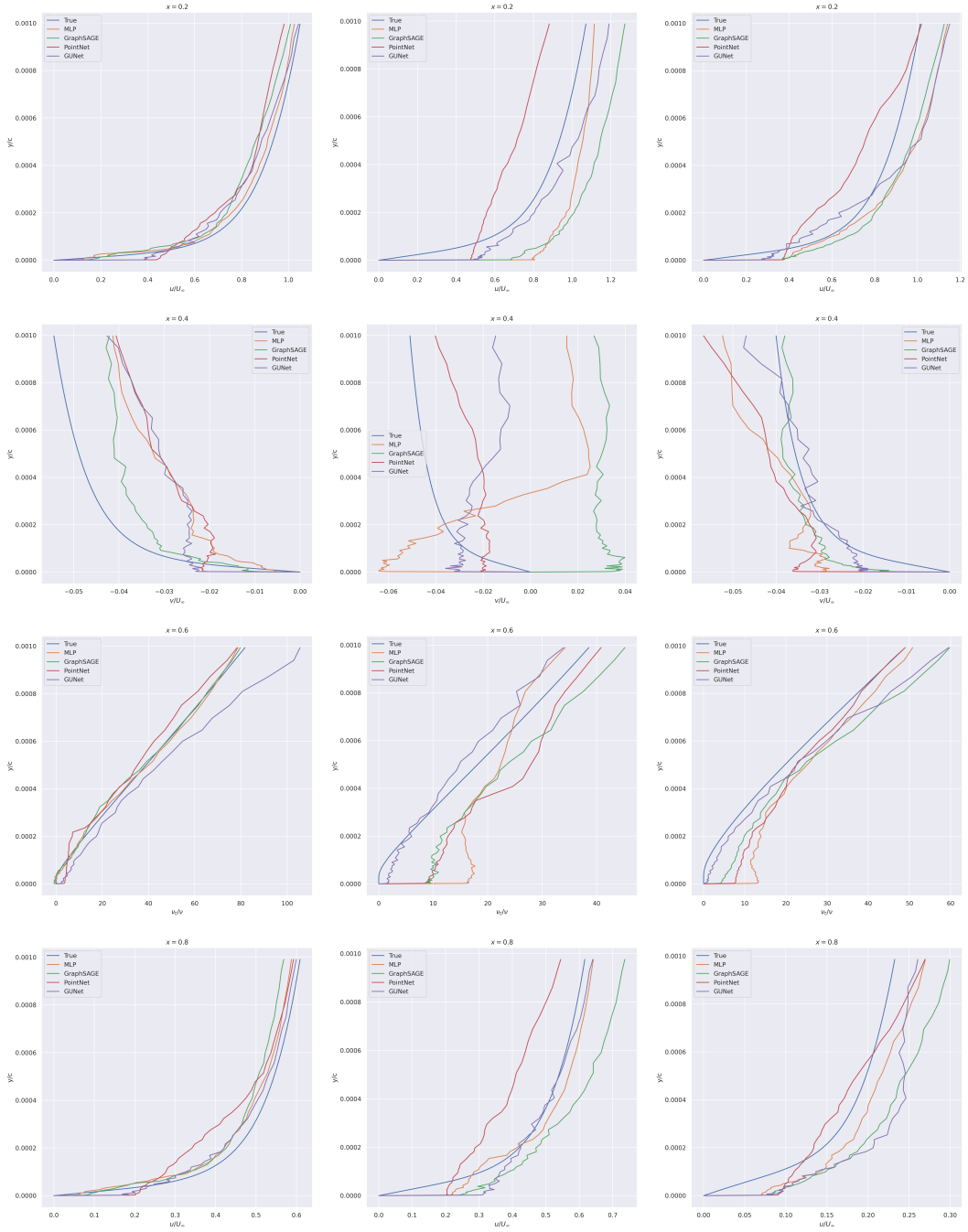
Figure 18: Comparison of the predicted boundary layers profiles on three random test geometries at different abscissas in the Reynolds extrapolation regime with respect to the true ones. Each column of plots represent a different airfoil and each line of plots represent a different abscissas. The x and y component of the velocity are denoted by u and v respectively and the turbulent viscosity is denoted by ν_t. Each quantity is normalized either by u_∞ the inlet velocity magnitude or ν the fluid viscosity.

40
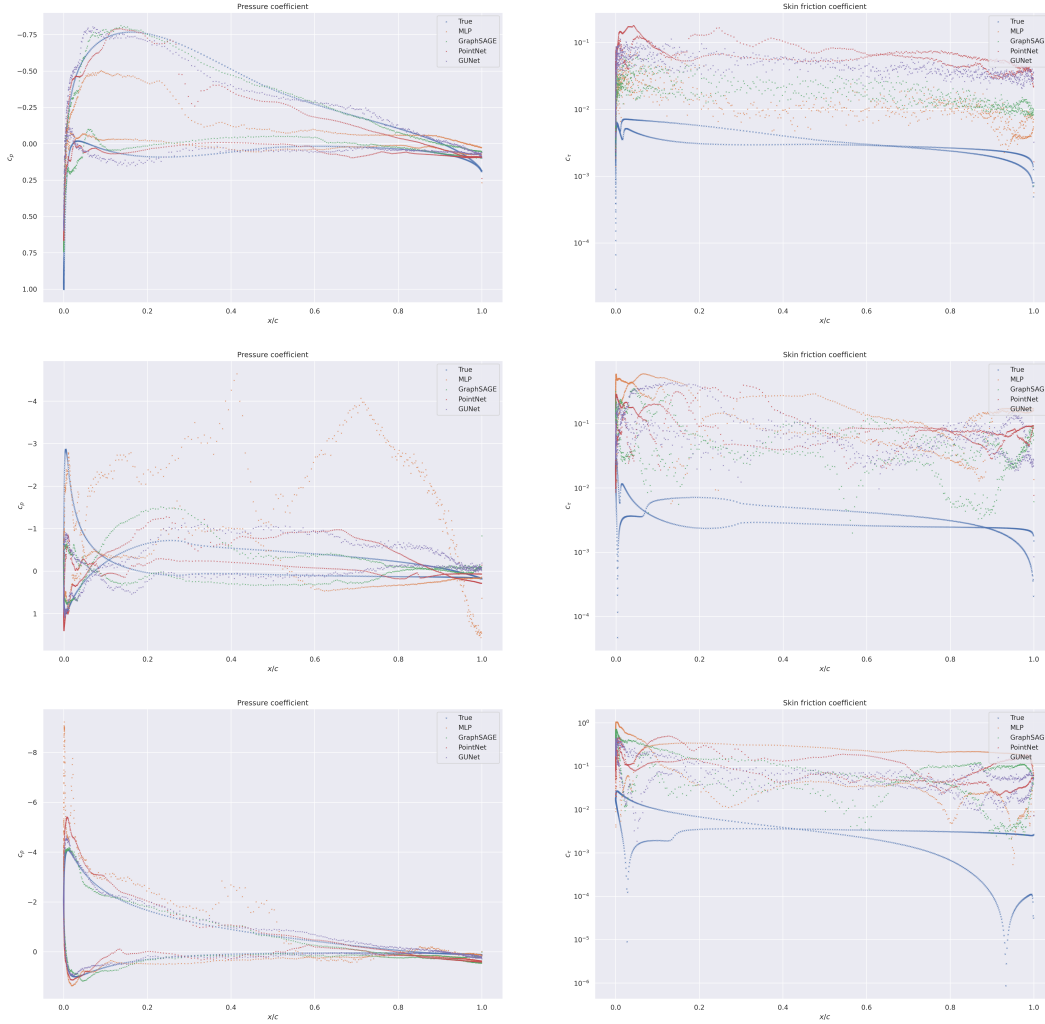
Figure 19: Comparison of the predicted surface coefficients profiles on three random test geometries in the Reynolds extrapolation regime with respect to the true one. (left) Surface coefficient c_p (right) Skin friction coefficient c_τ. Each line of plots represents a different airfoil. Skin friction coefficient plots are given in log scale.

Table 15: Relative errors (Spearman's rank correlation) for the predicted drag coefficient C_D (ρ_D) and lift coefficient C_L (ρ_L) in the angle of attack extrapolation regime. We want the Spearman's correlation to be close to one. Those quantities are computed as a post processing from the unnormalized regressed fields.

Model	Relative error		Spearman's correlation	
	C_D	C_L	ρ_D	ρ_L
MLP	8.00±0.85	1.06±0.29	0.04±0.17	0.861±0.049
GraphSAGE	**5.59±1.09**	0.82±0.30	0.05±0.17	0.908±0.019
PointNet	8.99±3.44	0.72±0.15	0.12±0.30	**0.936±0.022**
Graph U-Net	10.2±3.4	**0.69±0.14**	-0.20±0.13	0.934±0.022

41
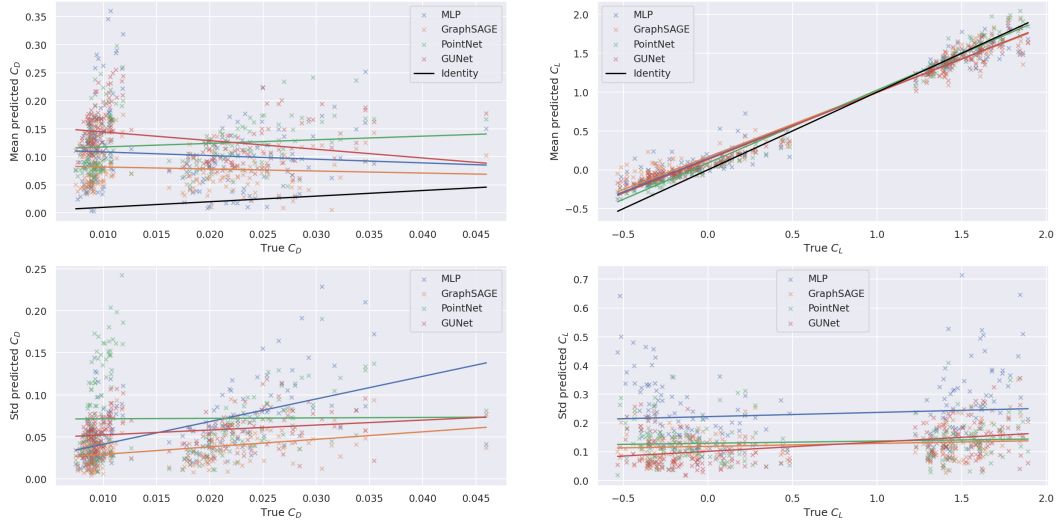
Figure 20: Predicted drag (left) and lift (right) coefficients with respect to the true ones in the angle of attack extrapolation regime. The mean (top) and standard deviation (bottom) of each point on the five copy of the trained models are separated for sake of readability. A linear regression is done for each point cloud in order to highlight linear trends. On the top plots, the Identity graph is given in black for comparison.

Table 16: Comparison of the mean squared error on the normalized fields for an MLP and standard GDL baselines on the different task for the associated test set. Only the reduced pressure is given on the surface as the other quantities are null via the boundary conditions. Those quantities are directly regressed by the models. The field denoted by $\bar{p}_s$ is the mean field reduced pressure at the surface of airfoils.

Field	Model	Task			
		Full	Scarce	Reynolds	AoA
$\bar{u}_x$ ($\times 10^{-2}$)	MLP	0.949 ± 0.058	1.647 ± 0.032	9.505 ± 1.275	6.965 ± 0.545
	GraphSAGE	$\mathbf{0.832 \pm 0.015}$	$\mathbf{1.457 \pm 0.125}$	$\mathbf{7.558 \pm 1.046}$	$\mathbf{4.435 \pm 0.334}$
	PointNet	3.500 ± 1.044	3.111 ± 0.303	9.422 ± 1.082	8.680 ± 2.337
	GUNet	1.517 ± 0.343	1.749 ± 0.190	8.383 ± 1.815	5.689 ± 0.708
$\bar{u}_y$ ($\times 10^{-2}$)	MLP	$\mathbf{0.978 \pm 0.172}$	$\mathbf{1.451 \pm 0.071}$	4.924 ± 0.800	10.630 ± 1.651
	GraphSAGE	0.994 ± 0.052	1.454 ± 0.123	$\mathbf{3.498 \pm 0.613}$	$\mathbf{9.400 \pm 2.167}$
	PointNet	3.645 ± 1.261	2.776 ± 0.395	7.129 ± 0.801	15.796 ± 5.392
	GUNet	2.028 ± 0.391	1.825 ± 0.182	5.250 ± 1.362	10.342 ± 2.788
$\bar{p}$ ($\times 10^{-2}$)	MLP	0.737 ± 0.131	3.904 ± 0.570	4.300 ± 0.188	11.711 ± 2.518
	GraphSAGE	0.661 ± 0.050	4.696 ± 0.804	$\mathbf{3.826 \pm 0.248}$	$\mathbf{10.908 \pm 2.164}$
	PointNet	1.151 ± 0.230	$\mathbf{3.294 \pm 1.052}$	4.011 ± 0.744	16.237 ± 6.601
	GUNet	$\mathbf{0.657 \pm 0.080}$	3.388 ± 0.844	4.483 ± 0.401	14.887 ± 7.502
ν_t ($\times 10^{-1}$)	MLP	0.190 ± 0.010	0.501 ± 0.076	1.312 ± 0.344	5.433 ± 0.237
	GraphSAGE	0.160 ± 0.021	0.611 ± 0.079	1.694 ± 0.383	5.178 ± 0.365
	PointNet	0.292 ± 0.048	0.558 ± 0.236	$\mathbf{1.273 \pm 0.443}$	$\mathbf{4.632 \pm 0.398}$
	GUNet	$\mathbf{0.146 \pm 0.014}$	$\mathbf{0.433 \pm 0.100}$	1.283 ± 0.310	5.348 ± 0.589
$\bar{p}_s$ ($\times 10^{-1}$)	MLP	1.130 ± 0.141	2.192 ± 0.529	20.898 ± 35.537	8.762 ± 1.589
	GraphSAGE	0.662 ± 0.103	1.945 ± 0.336	$\mathbf{1.797 \pm 0.338}$	7.638 ± 0.945
	PointNet	0.925 ± 0.259	1.827 ± 0.413	2.013 ± 0.758	$\mathbf{5.846 \pm 0.361}$
	GUNet	$\mathbf{0.386 \pm 0.071}$	$\mathbf{1.473 \pm 0.347}$	2.059 ± 0.442	6.967 ± 2.317

42
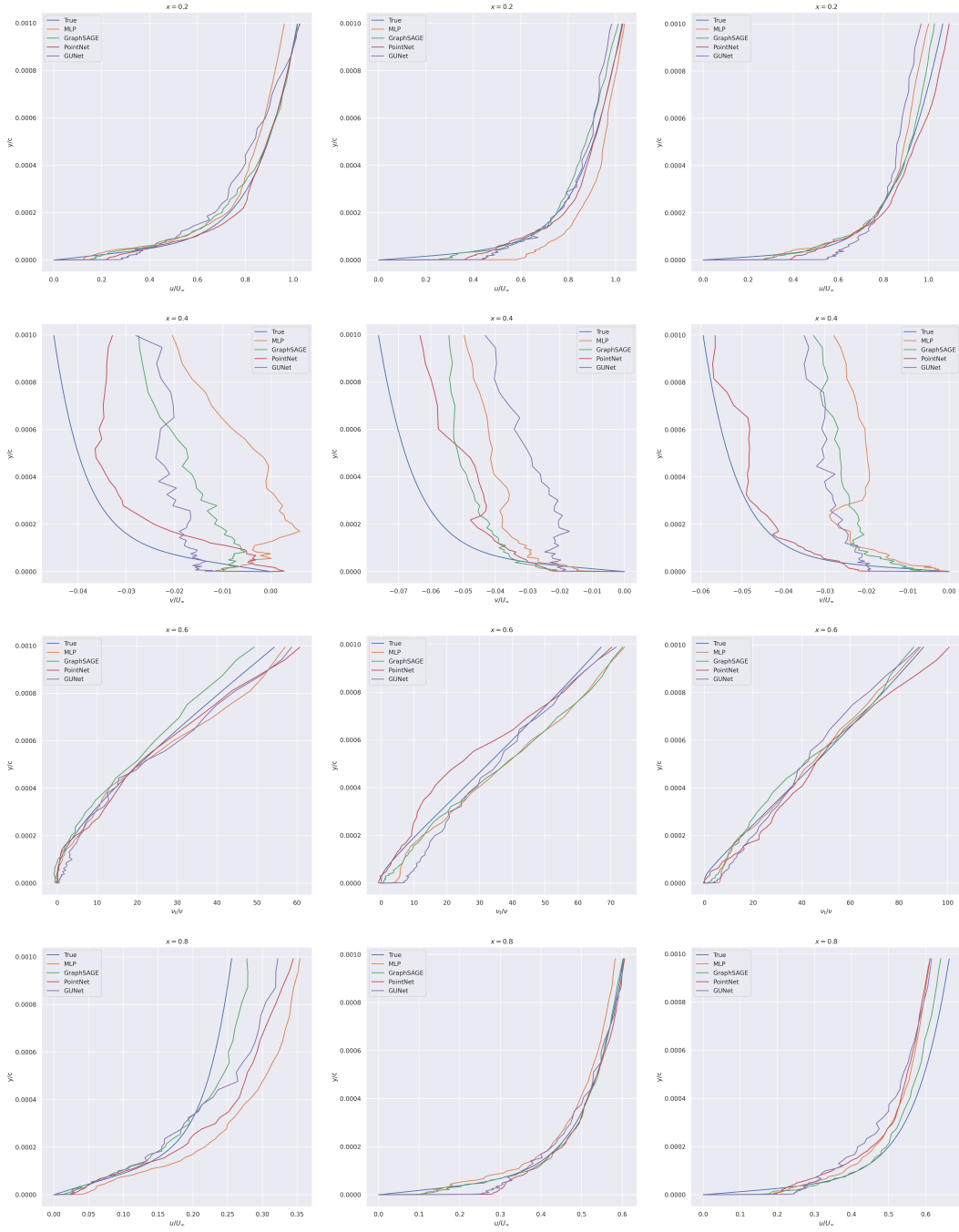
Figure 21: Comparison of the predicted boundary layers profiles on three random test geometries at different abscissas in the angle of attack extrapolation regime with respect to the true ones. Each column of plots represent a different airfoil and each line of plots represent a different abscissas. The x and y component of the velocity are denoted by u and v respectively and the turbulent viscosity is denoted by ν_t. Each quantity is normalized either by u_∞ the inlet velocity magnitude or ν the fluid viscosity.

43
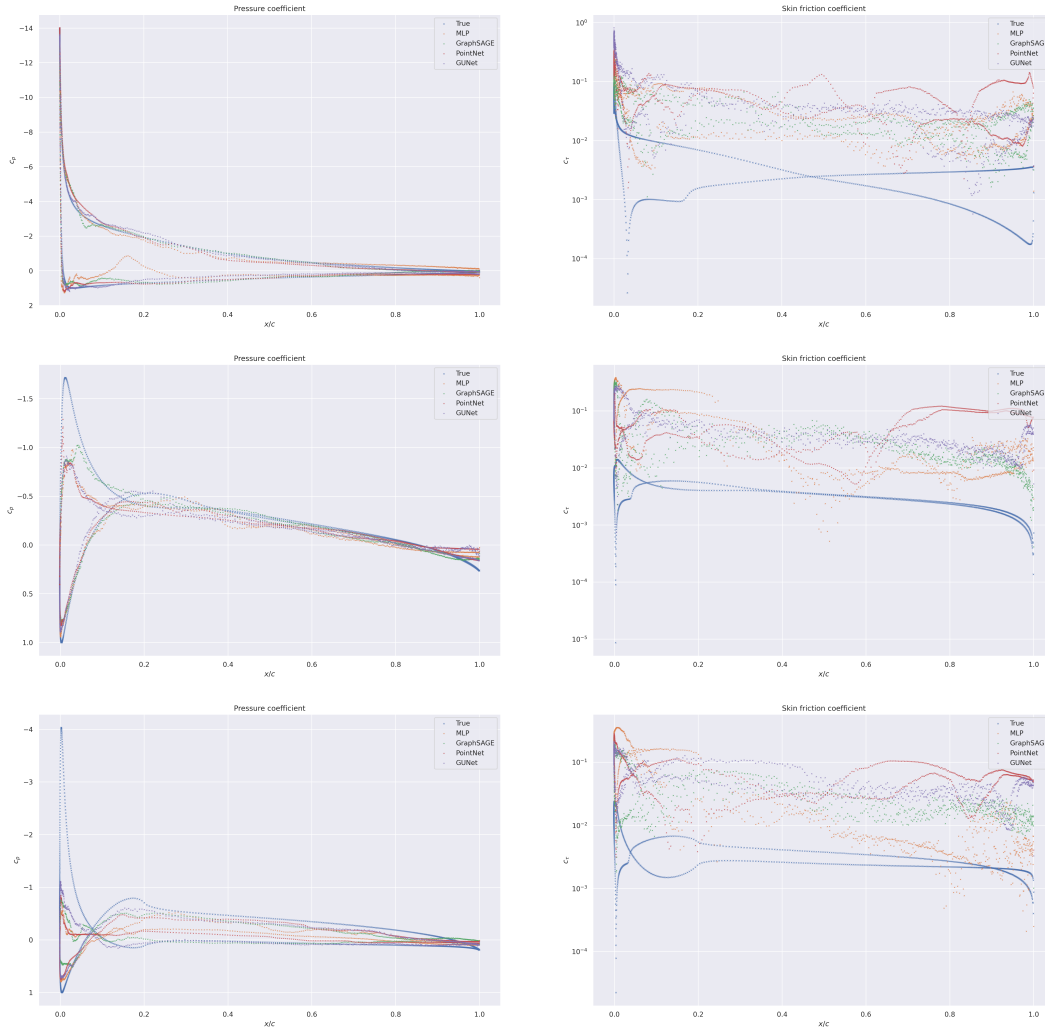
Figure 22: Comparison of the predicted surface coefficients profiles on three random test geometries in the angle of attack extrapolation regime with respect to the true one. (left) Surface coefficient c_p (right) Skin friction coefficient c_τ. Each line of plots represents a different airfoil. Skin friction coefficient plots are given in log scale.

44

Table 17: Comparison of the relative errors (Spearman's rank correlation) for the predicted drag coefficient C_D (ρ_D) and lift coefficient C_L (ρ_L) on the four different task for the associated test set. We want the Spearman's correlation to be close to one. Those quantities are computed as a post processing from the unnormalized regressed fields.

Field	Model	Task			
		Full	Scarce	Reynolds	AoA
C_D	MLP	4.289 ± 0.679	$\mathbf{2.950 \pm 0.144}$	13.397 ± 7.154	8.003 ± 0.848
	GraphSAGE	$\mathbf{4.050 \pm 0.704}$	3.504 ± 0.998	$\mathbf{8.971 \pm 1.278}$	$\mathbf{5.589 \pm 1.090}$
	PointNet	14.637 ± 3.668	8.350 ± 1.387	11.558 ± 2.783	8.991 ± 3.436
	GUNet	10.385 ± 1.895	6.871 ± 1.801	13.268 ± 1.818	10.238 ± 3.394
C_L	MLP	0.769 ± 0.108	0.662 ± 0.161	3.330 ± 3.287	1.061 ± 0.288
	GraphSAGE	$\mathbf{0.517 \pm 0.162}$	$\mathbf{0.385 \pm 0.097}$	$\mathbf{0.616 \pm 0.124}$	0.818 ± 0.300
	PointNet	0.742 ± 0.186	0.587 ± 0.135	0.897 ± 0.326	0.716 ± 0.145
	GUNet	0.489 ± 0.105	0.418 ± 0.129	0.868 ± 0.369	$\mathbf{0.693 \pm 0.136}$
ρ_D	MLP	-0.117 ± 0.256	-0.242 ± 0.078	-0.146 ± 0.153	0.038 ± 0.174
	GraphSAGE	-0.303 ± 0.124	-0.139 ± 0.175	0.013 ± 0.064	0.055 ± 0.171
	PointNet	-0.022 ± 0.097	-0.050 ± 0.272	0.006 ± 0.241	0.122 ± 0.300
	GUNet	-0.138 ± 0.258	-0.095 ± 0.232	0.028 ± 0.116	-0.195 ± 0.134
ρ_L	MLP	0.913 ± 0.018	0.923 ± 0.026	0.642 ± 0.274	0.861 ± 0.049
	GraphSAGE	0.965 ± 0.011	$\mathbf{0.981 \pm 0.006}$	$\mathbf{0.927 \pm 0.027}$	0.908 ± 0.019
	PointNet	0.938 ± 0.023	0.949 ± 0.019	0.898 ± 0.056	$\mathbf{0.936 \pm 0.022}$
	GUNet	$\mathbf{0.967 \pm 0.019}$	0.976 ± 0.009	0.904 ± 0.064	0.934 ± 0.022

45