Ref. Ares(2021)2930074 - 03/05/2021

upscale

Upscaling **P**roduct development **S**imulation **C**apabilities exploiting **A**rtificial inte**L**ligence for **E**lectrified vehicles

D2.3 Assessment of reduced order models for aerodynamic performance prediction

Authors

Nikolaos Kyriazis, Eugene De Villiers – Engys

Markus Mrosek , Carsten Othmer – VW

Luca Miretti – CRF

Enric Aramburu, Bhanu Prakash, Charalampos Tsimis, Albert Rodriguez de Liebana – IDIADA

Patrik Bangert – Algorithmica Technologies

Guglielmo Minelli, Per Hamlin – Volvo Cars

03/2021

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 824306.

Project Details

PROJECT TITLE	Upscaling product development simulation capabilities exploiting artificial intelligence for electrified vehicles
PROJECT ACRONYM	Upscale
GRANT AGREEMENT NUMBER	824306
INSTRUMENT	RIA
CALL	LC-GV-2018
STARTING DATE OF THE PROJECT	November, 1ST 2018
PROJECT DURATION	42 Months

The UpScale Project

The UPSCALE (Upscaling Product development Simulation Capabilities exploiting Artificial inteLligence for Electrified vehicles) goal is demonstrating the feasibility of using AI enhanced CAE methods in EV development processes, such as vehicle aerodynamics, battery thermal modelling and crash simulation and leading the deployment of AI tools for other CAE applications. UPSCALE is the first EU-project that has the specific goal to integrate artificial intelligence (AI) methods directly into traditional physics-based Computer Aided Engineering (CAE)-software and –methods. These CAE-tools are currently being used to develop road transportation not only in Europe but worldwide. The current focus of the project is to apply AI-methods to reduce the development time and increase the performance of electric vehicles (EVs) which are required by the automotive industry to reduce global emission levels. High performance computing (HPC) and CAE-software and –methods play a decisive role in vehicle development process. In order to make a significant impact on the development process, the two most HPC intensive CAE-applications have been chosen as use cases for the project: vehicle aero/thermal- and crashmodelling. When considering total automotive HPC usage, approximately 20% is used for aero/thermal simulations and up to 50% of HPC resources are utilized for crash simulations. By improving the effectiveness of these two areas, great increases in efficiency will lead to a 20% reduction of product time to market. Other novel modelling approaches such as reduced order modelling will be coupled to the AI improved CAE-software and -methods to further reduce simulation time and ease the application of optimization tools needed to improve product quality. Through the combined effort of universities, research laboratories, European automotive OEMs, software companies and an AI-SME specialized in machine learning (ML), the UPSCALE project will provide a unique and effective environment to produce novel AI-based CAE-software solutions to improve European automotive competeviness.

The UpScale Consortium

PARTICIPANT Nº	PARTICIPANT ORGANISATION NAME	COUNTRY
1 (Coordinator)	**IDIADA AUTOMOTIVE TECHNOLOGY SA (IDIADA),**	Spain
2	VOLVO PERSONVAGNAR AB (Volvo Cars)	Sweden
3	VOLKSWAGEN AG (VW)	Germany
4	CENTRO RICERCHE FIAT SCPA (CRF)	Italy
5	ESI GROUP (ESI GROUP)	France
6	ENGYS LTD (ENGYS LTD)	United Kingdom
7	Kompetenzzentrum - Das Virtuelle Fahrzeug, Forschungsgesellschaft mbH (VIF)	Austria
8	VRIJE UNIVERSITEIT BRUSSEL (VUB)	Belgium
9	ECOLE NATIONALE SUPERIEURE D'ARTS ET METIERS (ENSAM PARISTECH)	France
10	ALGORITHMICA TECHNOLOGIES GMBH (ALGORITHMICA)	Germany
11	F INICIATIVAS I MAS D MAS I SL (F-INICIATIVAS)	Spain

Document Details

DELIVERABLE TYPE	Report
DELIVERABLE Nº	2.3
DELIVERABLE TITLE	Reduced order models for aerodynamic performance prediction
NAME OF LEAD PARTNERS FOR THIS DELIVERABLE	IDIADA
VERSION	1
CONTRACTUAL DELIVERY DATE	M30
ACTUAL DELIVERY DATE	
DISSEMINATION LEVEL	Public

Revision History

The following table describes the main changes done in the document since it was created

REVISION	DATE	DESCRIPTION	AUTHOR (ORGANIZATION)
V.0	08/03/2021	Contribution to the document	Markus Mrosek, Carsten Othmer (VW)
V.0	08/03/2021	Contribution to the document	Luca Miretti (CRF)
V.0	09/03/2021	First complete draft of the document	Nikolaos Kyriazis (ENG)

V.1	25/03/2021	Task leader revision	Enric Aramburu (IDIA)
V.2	26/03/2021	WP leader revision	Luca Miretti (CRF)
	30/04/2021	Project coordinator	Albert R.de Liébana (IDIADA)

Disclaimer

Table of contents

1. Executive Summary

This report is a part of work package 2, focusing on AI based design for Aerodynamics, and describes the work performed in Task 2.3. The results will be used for the choice of the most promising reduced order model to be used for relevant use cases, such as the aerodynamic optimization of actual vehicles in WP4.

The main goal of Task 2.3 is to identify the most suitable reduced order model approach, among the ones developed in Task 2.2, to be applied for the aerodynamic optimization in a relevant industrial environment.

First of all the accuracy, performance and potentiality of the ROMs developed in Task 2.2, where POD-I and ML approaches were tested on a simplified 2D vehicle, is compared and the most promising one is chosen for subsequent tests and developments. The identified workflow is then applied to "higher fidelity" datasets, to ensure that the selection of reduced order modelling technique is valid for relevant use cases. The approach here used for the validation is the application of the methodology on a complete 3D electrified vehicle. In order to ensure homogeneity of results with previous steps of the activity the 3D DrivAer model is used. Following the prescribed workflow, a dataset of about 1000 different geometries is generated starting modifying the baseline shape, flow fields and aerodynamic coefficients are evaluated by means of CFD simulations and the neural networks are trained with these input data. Finally, capabilities and accuracy of the trained nets are evaluated both in terms of drag coefficient and flow fields prediction.

This deliverable doesn't deviate from the plan in regard to its content or delivery date.

2. Methods comparison

2.1. Training dataset

In order to perform a fair assessment of accuracy and performance between the different reduced order models developed in the previous tasks, both “deterministic” and “non-deterministic” models are tested on the same dataset, and obtained results are compared.

The dataset chosen for this validation are the “ClosedAGS” and “OpenAGS” versions of the 2D electrified DrivAer [1], consisting each of one thousand different shapes and CFD solutions generated on the bi-dimensional simplified shape. Full details on the datasets can be found in D2.2: Reduced order models for aerodynamic performance prediction. A quick reminder on the adopted database is provided in this paragraph.

First of all a DOE of shapes was generated starting from the original model by means of surface morphing, thirteen different geometrical parameters were used. An example of the different shapes available in the dataset is shown in Figure 1.

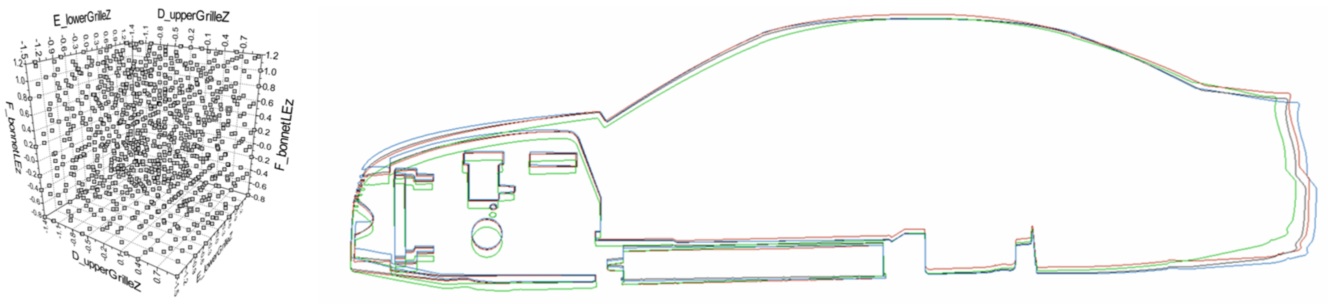


Figure 1: Parameters distribution and example of different shapes available in the 2D DrivAer “ClosedAGS” dataset

For each generated shape a CFD simulation was performed, and results were extracted in terms of drag coefficients and flow fields, in this case resampled on a uniform grid of 256x256x1 nodes. These outputs were used to train the POD-I and the NN reduced order models, subjects of the assessment presented in this chapter.

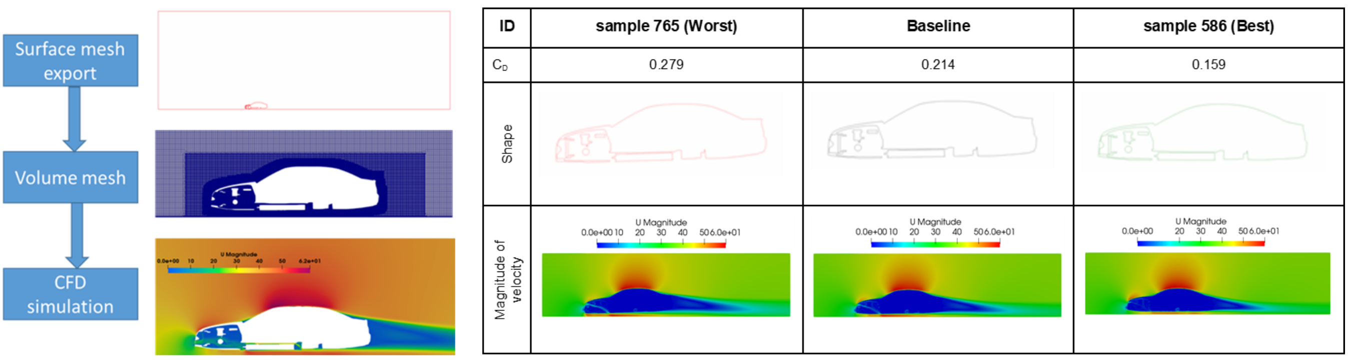


Figure 2: CFD workflow and example of results on some shapes available in the DOE

2.2. Drag coefficient prediction

The comparison between a deterministic interpolation-based method and the non-deterministic Neural-Network-based approach (NN) for predicting the drag coefficient is demonstrated in this section. The “pure” open and closed AGS data sets (and not the mixed ones) have been utilized, as they were found from previous investigations in D2.2 to provide the most accurate models. The different databases as well as the methodologies have been described in sections 2.1, 3.1 and 3.2 of the D2.2 report, respectively.

In the case of the interpolation model, we make use of the TPS (Thin-Plate-Spline) approach that proved successful for the very similar task of interpolating the POD base coefficients in D2.2. Training and prediction were performed on a standard CPU of a standard desktop computer.

For the NN, the Adagrad optimizer and 5 convolution blocks were used. More details regarding the NN training configurations can be found in Table 3 of D2.2. The training was run in Cirrus HPC, on one NVIDIA Tesla V-100-SXM2-16GB (Volta) GPU accelerator out of four which exist in each GPU node, while the prediction was performed on a Quadro P2000 5GB GPU accelerator in a desktop computer.

The drag coefficient mean error has been evaluated for the test data set, which is a subset of each one of the two previously mentioned databases. It has to be noted here that the samples of the test data set have not been used during the training of the model. For a consistent comparison between TPS-interpolation and NN, the random splitting of the samples into test and train data sets is kept the same for the two methodologies, both for the open and the closed database.

The mean error for the drag prediction in the test data set, its standard deviation, and the training and prediction wall clock times are shown in . The mean error expression of Eq. 1 of D2.2 is followed here, but the sum now is divided by the number of samples:

$$D_{error} = \frac{1}{N}\sum_{s=1}^{N}\frac{| D_s^{AI} - D_s^{CFD} |}{| D_s^{CFD} |} \qquad \text{(Eq. 1)}$$

As can be seen in , the deterministic TPS-based approach outperforms the NN in terms of c_d-prediction error by a factor of three to four. The TPS training times are smaller than their NN counterparts by a factor of about 5000. Also the prediction via TPS is significantly faster than with an NN, but for both methods the absolute prediction time values are small enough to provide the desired real-time capability.

The error for the open data set is higher than the one for the closed data set. For the NN, one possible explanation is that the open or closed status of the grille shutters cannot be properly captured with the given sampling resolution as their size is comparable to the sampling step.

For the OpenAGS database, Figure 3 shows the histograms of the drag coefficient. The inferiority of the NN w.r.t. the TPS approach is also reflected in the error distribution. The maximum NN error noticed in the open database was found to be 18.35%. Still, more than two thirds of the test cases in the open database have errors below or equal to the database mean error, and in general, the qualitative trend in the drag coefficient was captured correctly by the NN, as can be deduced from the correlation plot in Figure 4

Figure 4. As a result, despite performing worse than the deterministic method, the NN-based classification algorithm demonstrated the capability to learn the geometric dependence of the drag coefficient and to predict it with a still acceptable mean error of 2-3%.

Table 1: Comparison between TPS and Neural Network for drag coefficient c_d prediction. The prediction errors (Eq. 1) are given as mean +/- standard deviation over the test set, and the training and prediction (single test sample) wall clock times are reported.

Database	Method	Error c_d [%]	Training time [s]	Prediction time [s]
OpenAGS	Neural Network	3.10 +/- 2.78	462.6	0.0317
	TPS	0.95 +/- 1.06	0.10	0.0004
ClosedAGS	Neural Network	2.38 +/- 2.04	504.6	0.0231
	TPS	0.63 +/- 0.60	0.10	0.0004

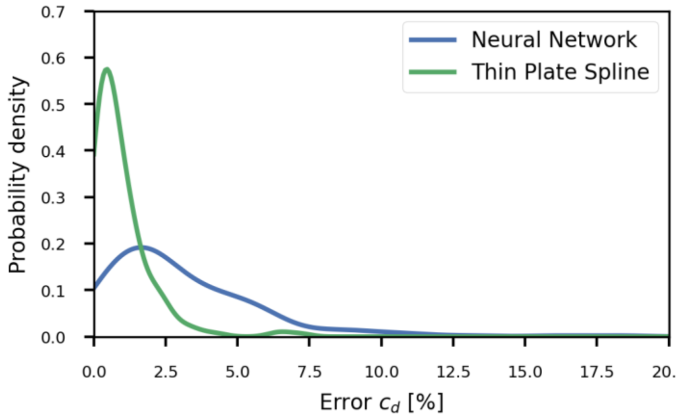


Figure 3: Distributions (kernel density estimation) of the drag coefficient prediction errors for the 201 test samples of the OpenAGS dataset.

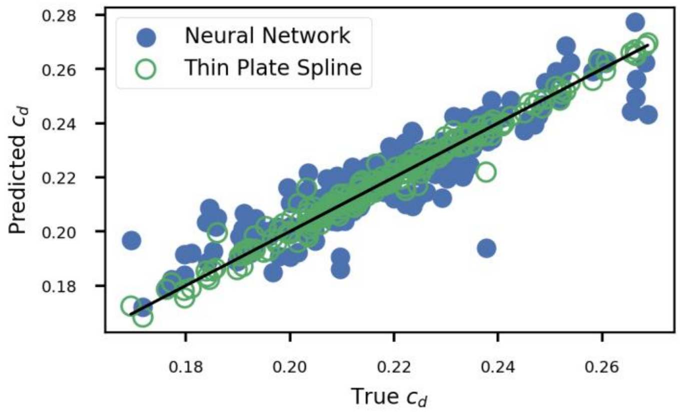


Figure 4: Correlation between the predicted drag coefficient values by either Neural Network or Thin Plate Spline with the true values from CFD (OpenAGS dataset). For perfect predictions, the points would lie on the black diagonal line.

2.3. Flow fields prediction

In this section, the accuracy and the efficiency of the deterministic POD+I (Proper Orthogonal Decomposition + Interpolation) and the non-deterministic NN-based method for predicting the flow fields will be assessed. The previously mentioned data sets and their subsets have been used for the flow prediction as well. The methodologies have been described in sections 3.1 and 3.2 of D2.2 in detail, while the NN training configurations can be found in Table 4 of the same report.

Regarding the NN approach, the Adam optimizer and 7 convolution layers have been used. For POD+I we employed TPS again, and the number of used POD modes were not changed: 88 for *U* and 157 for *p*. Model training and prediction were performed on the same hardware as for the drag prediction (see previous section). The mean errors of the flow fields in the test data set are evaluated with the same metrics as in D2.2:

$$p_{error} = \sum_{s=1}^{N} \left(\sum_{i=1}^{P} \frac{| p_{i,s}^{AI} - p_{i,s}^{CFD} |}{| \mathbf{U}_\infty |^2} \right) \Big/ NP \qquad \text{(Eq. 2)}$$

$$U_{error} = \sum_{s=1}^{N} \left(\sum_{i=1}^{P} \frac{| \mathbf{u}_{i,s}^{AI} - \mathbf{u}_{i,s}^{CFD} |}{| \mathbf{U}_\infty |} \right) \Big/ NP \qquad \text{(Eq. 3)}$$

presents a comparison between NN and POD+I for the prediction errors of velocity and pressure as well as training and prediction times, and Figure 5 compares the error distributions for both methods applied to the OpenAGS database. As opposed to the case of drag coefficient prediction (previous section), the NN can now compete with POD+I in terms of accuracy. It even outperforms POD+I for the velocity prediction while being only slightly worse for pressure. Also, the contrast in training time (factor 5000 for c_d) has decreased considerably to a mere factor of about 20.

As before, predictions for the OpenAGS database seem to be the harder task, with errors lying slightly above the ones for the closed database for either method and both fields, and, again, the poor representation of the open grille shutters may be the reason for this. But even for this database, the errors are well below 1%, meaning that both POD+I and NN serve as *quantitatively* reliable prediction tools for the volumetric flow fields.

Table 2: Comparison of the prediction errors for velocity and pressure as well as training and prediction times between Neural Network and POD+I. As separate models were created for pressure and velocity, the shown training and prediction times are the sum of the respective times for the two models.

Database	Method	Error velocity [%]	Error pressure [%]	Training time [s]	Prediction time [s]
OpenAGS	Neural Network	0.71 +/- 0.17	0.69 +/- 0.20	2999	0.016
	POD+I	0.75 +/- 0.16	0.49 +/- 0.11	135	0.11
ClosedAGS	Neural Network	0.60 +/- 0.11	0.47 +/- 0.10	2997	0.016
	POD+I	0.67 +/- 0.11	0.41 +/- 0.08	179	0.10

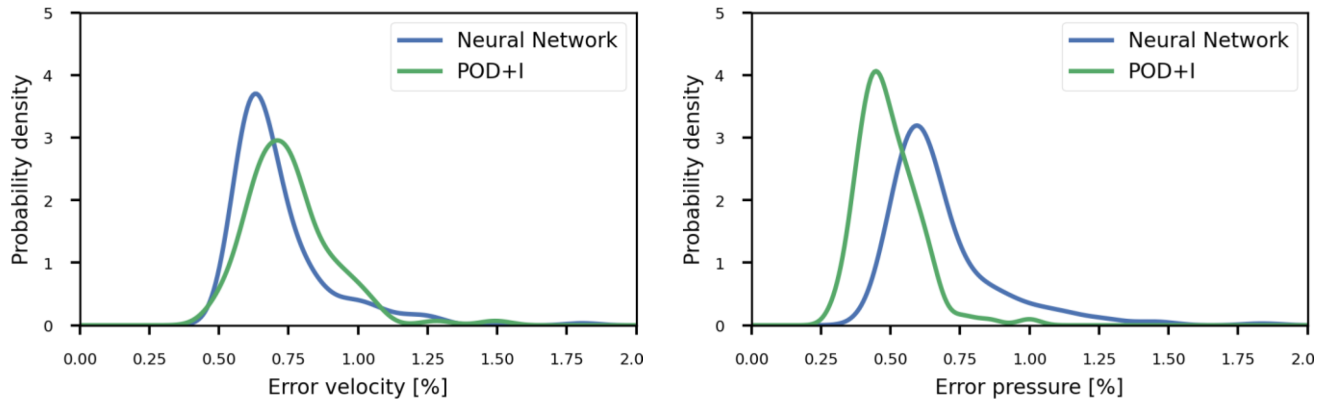


Figure 5: Comparison of the velocity (left) and pressure (right) prediction error distributions (kernel density estimations) for the 201 test samples of the OpenAGS dataset between Neural Network and POD+I.

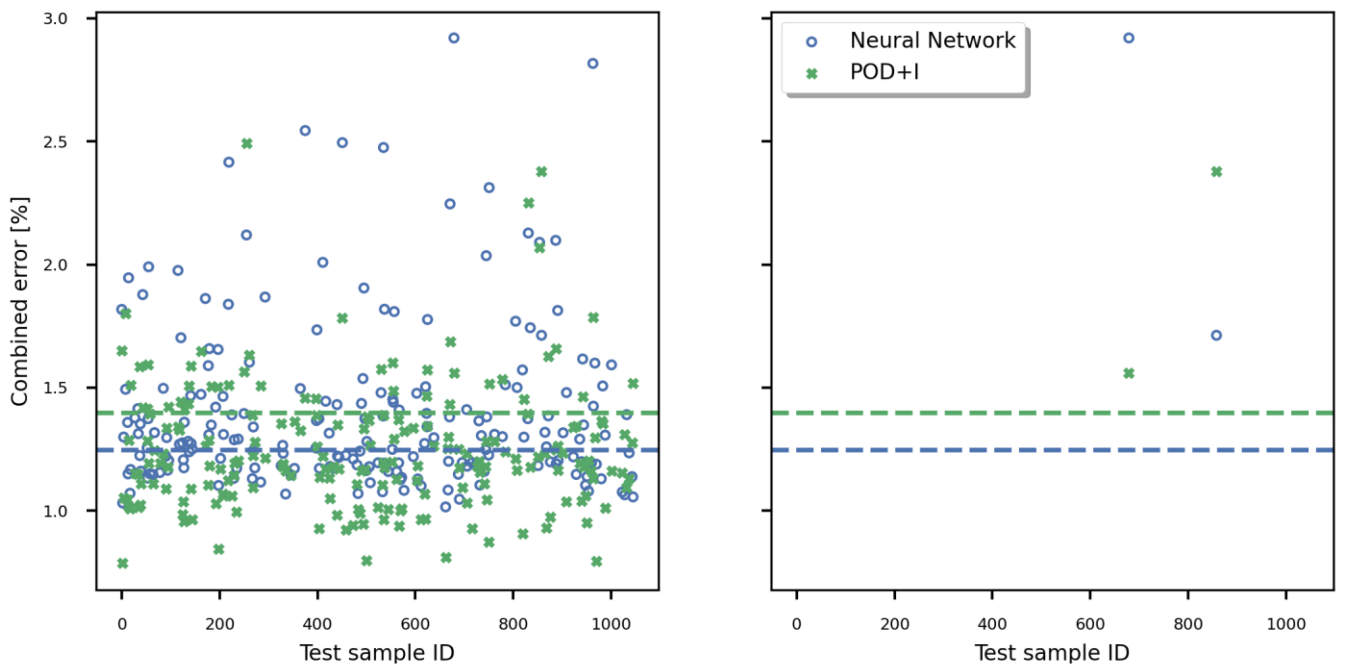


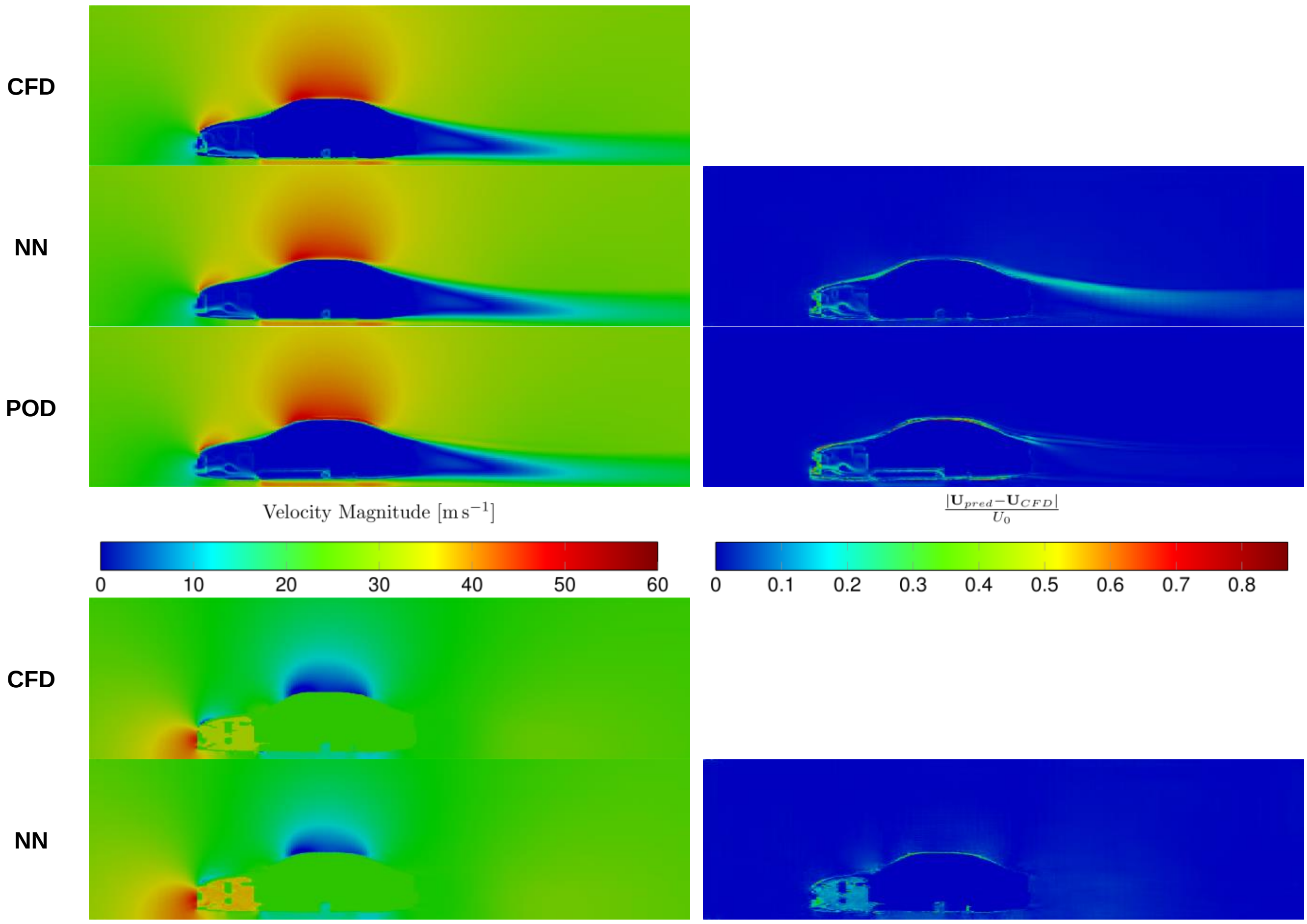
Figure 6: Combined errors (velocity error + pressure error) for the test samples of the OpenAGS dataset. Horizontal lines indicate the mean errors over all test samples. The left plot includes all individual test samples, whereas the right plot only shows the two test samples where POD+I performed best compared with the Neural Network (ID 679) and, analogously, where the Neural Network performed best compared with POD+I (ID 858). Those two samples were chosen for the qualitative comparison of the methods.

For a more *qualitative* assessment of their respective predictive capabilities we have created vis-à-vis comparisons between NN and POD+I for the flow fields of selected test cases. In order to appreciate any possible difference between NN and POD+I, we decided to focus on the more difficult OpenAGS database, and we selected the two test samples with the largest difference in the combined $U + p$ prediction error: one where the NN is most superior to POD+I, and one where POD+I maximally outperforms NN. Figure 6 shows the combined errors for all OpenAGS test samples (left) and the ones for the thus selected extreme samples (right). These are:

- Test sample ID 679
  - NN: 1.81 % error for U, 1.11 % for p → 2.92 % combined
  - POD+I: 0.94 % for U, 0.61 % for p → 1.55% combined
  - See Figure 7
- Test sample ID 858
  - NN: 1.08 % error for U, 0.63 % for p → 1.71 % combined
  - POD+I: 1.51 % for U, 0.87 % for p → 2.38 % combined
  - See Figure 8.

As can be seen from Figure 7 and Figure 8, for POD+I and NN alike, the errors in both velocity and pressure concentrate along the surface of the car, which is a direct consequence of the employed spatial sampling method not being body-fitted. An additional critical area is the engine bay, where both methods fail in predicting the right pressure level and the details of the velocity distribution in this geometrically complex environment, especially for sample ID 679. A higher spatial sampling resolution might alleviate these inaccuracies. For the velocity prediction, the shear layer of the flow leaving the top edge of the trunk seems to be another demanding area. Its position in the vertical direction is strongly geometry-dependent and it comes naturally with large gradients in this direction, which makes accurate predictions in this area particularly difficult.

Overall, even though we picked a test sample pair with the largest contrast in the prediction error between POD+I and NN, we can hardly discern individual weaknesses or strengths. In terms of the qualitative accuracy of the predicted fields, both methods perform satisfactorily and face similar challenges.



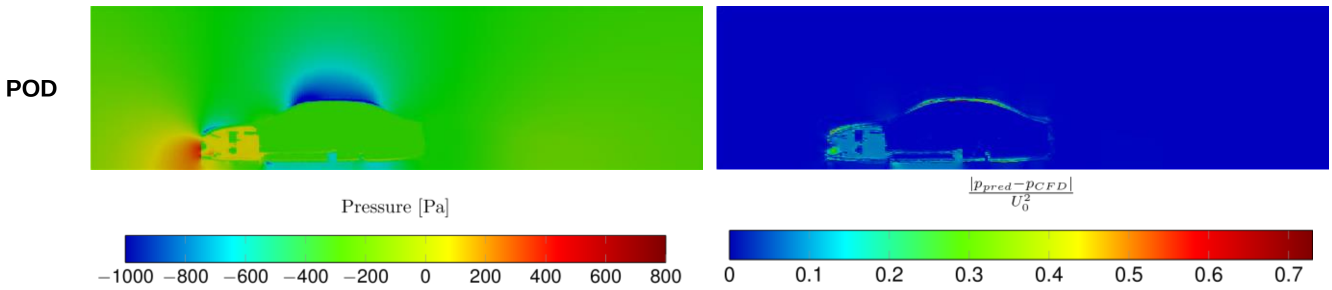


Figure 7: Comparison of fields from CFD with Neural Network and POD+I predictions for the test sample where POD+I performed best compared with the Neural Network (ID 679). Top: velocity, bottom: pressure, right: differences between predictions and CFD.

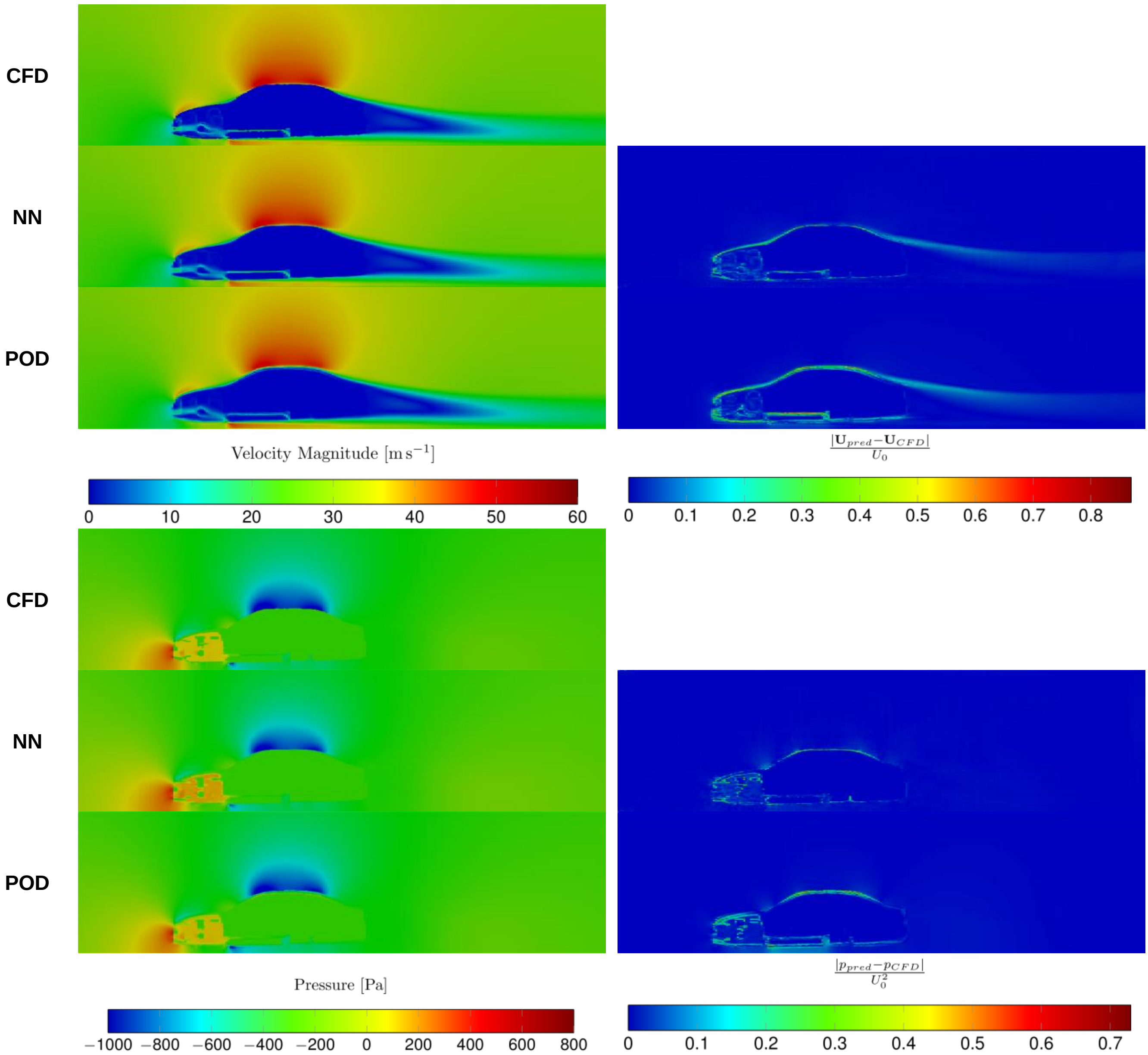


Figure 8: Same as Figure 7, but with the test sample where NN performed best compared with POD+I (ID 858).

3. ROM selection

3.1. Method selection for further investigations

From the assessment of the two ROM approaches in sections **¡Error! No se encuentra el origen de la referencia.** and **¡Error! No se encuentra el origen de la referencia.**, it can be inferred that the main advantage of the POD+I approach over the ML algorithms is reduced computational cost. Besides, POD+I seems to be three to four times more accurate than NN in the drag coefficient prediction. Regarding the error of the methods in the flow fields prediction, the NN approach produced lower error for the velocity but higher error for the pressure prediction. All in all, the difference in the pressure and velocity errors between the two methods is negligible.

On the other hand, the ML approach could lead to various benefits in terms of universality of the approach, as it can be potentially applied to predict the drag coefficient and the volumetric flow fields in geometries that are outside the database design space, as new features/physical parameters can be introduced that were not used during the training. An example of potential application is shown in section 3.2, where the already trained networks are used to predict drag coefficients and flow fields on "never-seen" modifications on the 2D DrivAer model. A preliminary assessment of the results obtainable with this approach is performed, showing that the accuracies, especially for the flow fields prediction, do not differ too much from the mean error values of the test data set that have been documented in the previous section; however, regions where the unseen modifications are geometrically located are the ones where the maximum errors are located. According to these initial results and despite of these deviations, this approach can be considered a point of interest for future investigations in the field.

Based on the above, the ML approach has been selected as the most suitable method to proceed in the UPSCALE project and it will be employed in the high fidelity application of section **¡Error! No se encuentra el origen de la referencia.**, as well as for predicting the drag coefficient and the flow fields of the vehicles in WP4.

3.2. Additional potentialities

One of the strengths of the ML-based approach, as mentioned, is the possibility to predict aerodynamic results on shapes characterized by new features/physical parameters not included in the training dataset, "detaching" the prediction capability from the original parametrization. This feature, compared to surrogate models based only on interpolation, can be considered a breaking point with respect to state-of-the-art techniques, opening several scenarios in industrial product development.

A test case is performed on the DrivAer 2D model, using the already trained networks to predict aerodynamic coefficients and flow fields on additional configurations generated ad-hoc for this application. These new configurations consist of "never-seen" modifications applied on the baseline shape, always obtained by surface morphing. Involved modifications span from simple variations (e.g. linear combination of training parameters) to completely new surface features (e.g. modifications on underbody components not included in the original DOE).

The full list of new generated configurations, with a draft description of the modification, is presented in Table 3.

Table 3: 2D DrivAer geometric shapes generated for the "never-seen" configurations test case

ID	Modified region	Description
1050	Front bumper	Baseline + front bumper translation -20mm
1051		Baseline + front bumper translation -100mm
1052	Underbody	Baseline + underbody spar -30mm
1053		Baseline + underbody spar -80mm
1054	Roof	Baseline + roof translation -70 mm
1055		Baseline + roof translation -20 mm
1056		Baseline + roof translation +20 mm
1057		Baseline + roof translation +70 mm
1058	Engine belly pan	Baseline + engine belly pan -100 mm
1059		Baseline + engine belly pan -20 mm
1060		Baseline + engine belly pan +20 mm
1061		Baseline + engine belly pan +100 mm
1062	Cowl	Baseline + cowl x-translation -100mm
1063		Baseline + cowl x-translation -20mm
1064		Baseline + cowl x-translation +20mm
1065		Baseline + cowl x-translation +100mm
1066	Trunk	Baseline + trunk rotation -10deg
1067		Baseline + trunk rotation -2deg
1068		Baseline + trunk rotation +2deg
1069		Baseline + trunk rotation +10deg
1070	All	Baseline + front bumper translation -80mm underbody spar -60mm roof translation -50 mm engine belly pan +70 mm cowl x-translation +70mm baseRotation -7deg
1071		Baseline + front bumper translation -30mm underbody spar -20mm roof translation +20 mm engine belly pan -30 mm cowl x-translation -20mm baseRotation +3deg

A visual example of some of the tested configurations is reported in Figure 9, Figure 10 and Figure 11.

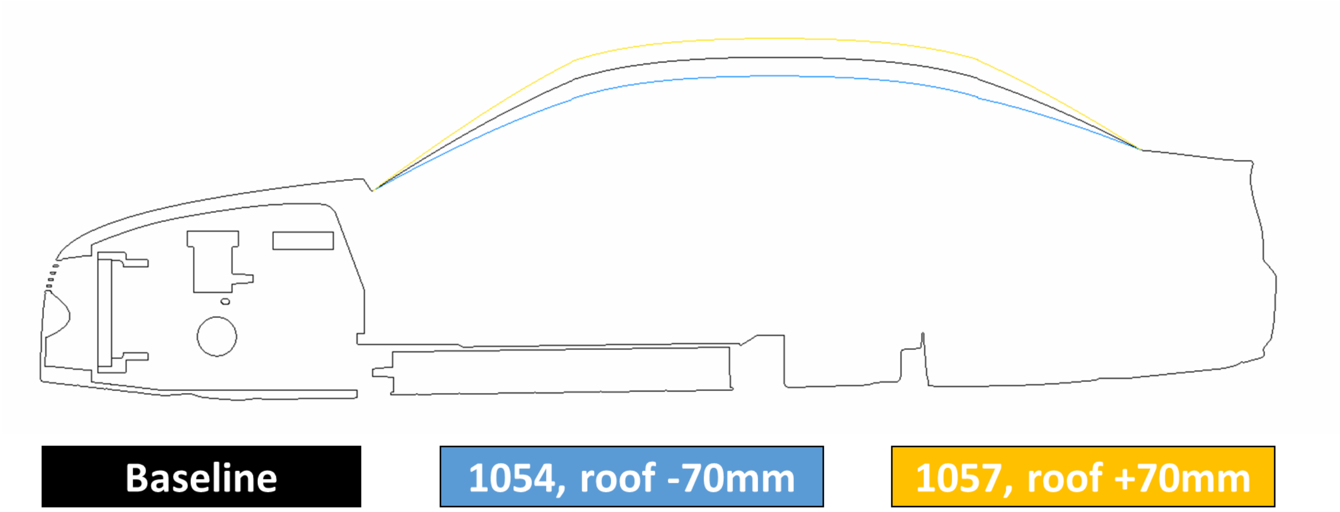


Figure 9: “Never-seen” modifications shapes, modified roof region

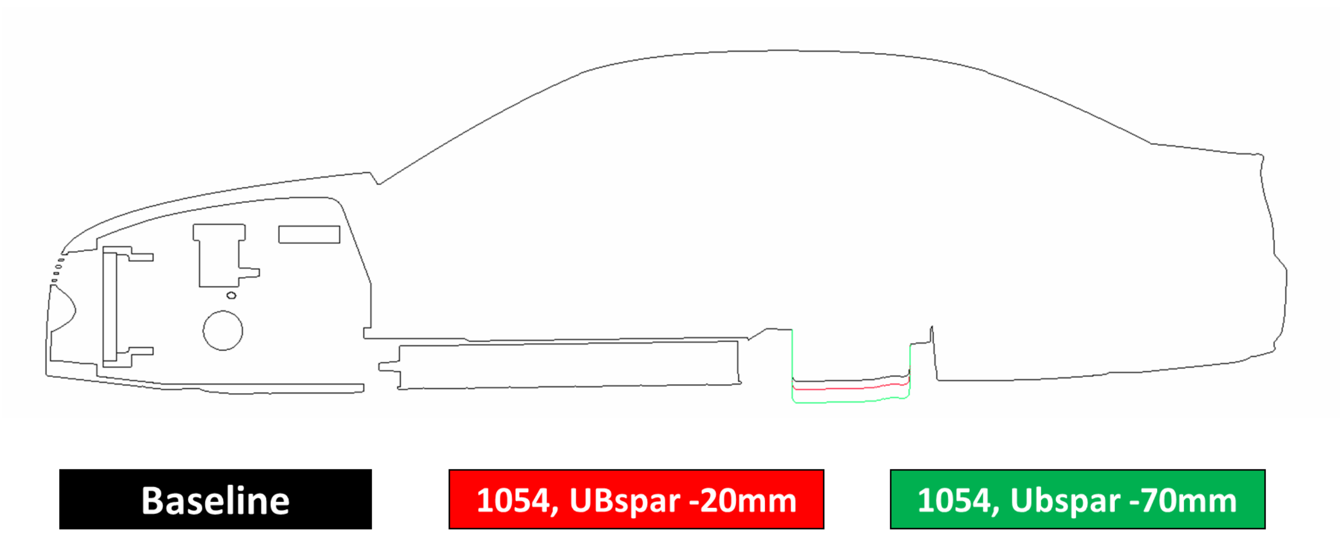


Figure 10: “Never-seen” modifications shapes, modified underbody region

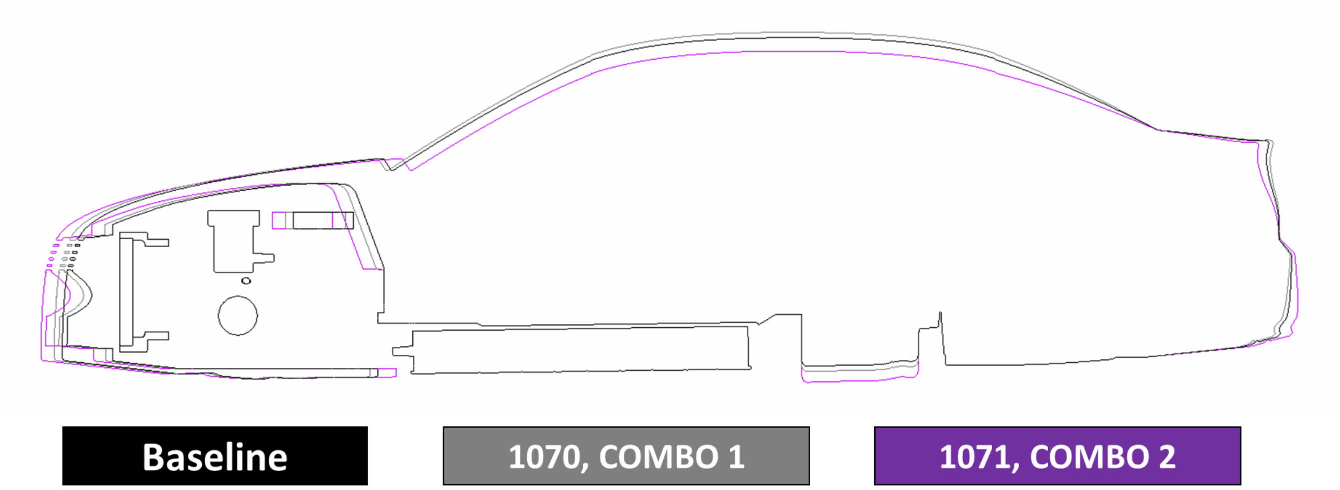


Figure 11: “Never-seen” modifications shapes, modified multiple regions

Among the presented shape variations, different difficulty levels exist for the network validation: roof modification can be considered a linear combination of two geometrical parameters used to train the database (windscreen angle and rear window angle, see Figure 12 or refer to D2.2); underbody spar modifications were totally not present in the original parameters dataset; combo shapes, obtained by merging all the never-seen parameters, can be considered the hardest test-case for the network.

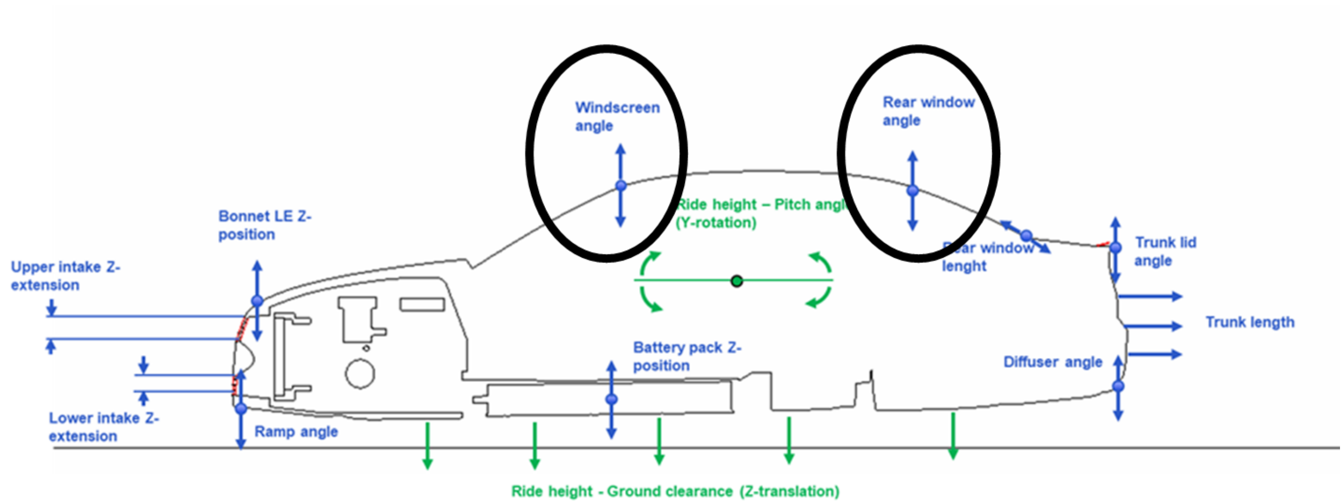


Figure 12: Parameters used to generate the dataset on which NN is trained, roof modifications presented here is a linear combination of two existing parameters, highlighted by circles

For all the twenty-one new generated shapes (unknown database), the original network is used to predict both drag coefficients and flow fields. As a matter of validation, full CFD workflow is applied on each shape, and these "high-fidelity" results obtained by simulation are compared with reduced order model prediction.

The mean error of the drag coefficient for the unknown database was found to be 2.71% and its standard deviation 1.77%. It can be inferred that the trained model works reasonably well even for the unknown cases, as the mean error remains in the same range, i.e. from 2% to 3%, and the worst prediction gives an error of 7.16% (see Figure 13). Nevertheless, the small number of samples prevents any further comments on the relatively low error value noticed in the worst-case scenario.

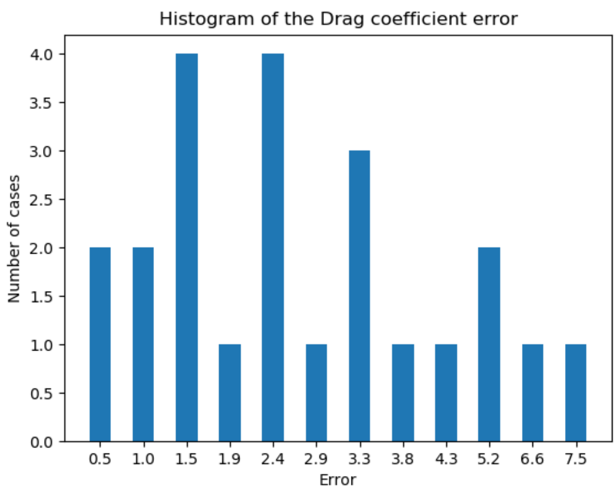


Figure 13: Histogram of the Drag coefficient error for the unknown database. Due to the small size of the database, the number of the samples is denoted in the y axis instead of the percentage of the samples.

The pressure and velocity fields, based on the closed AGS model, are predicted next. In Table 4 the corresponding mean errors and their standard deviations are shown, whereas in Figure 14 the histograms of the above-mentioned errors are depicted. The errors remain in the same range as the ones calculated for the closed test data set, but again, the small size of the unknown dataset prohibits any further deductions. As expected, the mean errors are significantly higher if the open AGS model is used, i.e. 1.58% for the pressure and 1.02% for the velocity magnitude fields. This is reasonable as the unknown cases have the lower grille shutters closed and therefore the most appropriate model for predicting the flow fields is the one generated by using the closed AGS data set.

Table 4: Error in the pressure and velocity magnitude prediction and the standard deviation of these errors for the unknown database.

Train database	**p** (error %)	**U** (error %)	**std(p_{error})**	**std(U_{error})**
closedAGS	0.695	0.891	0.370	0.372

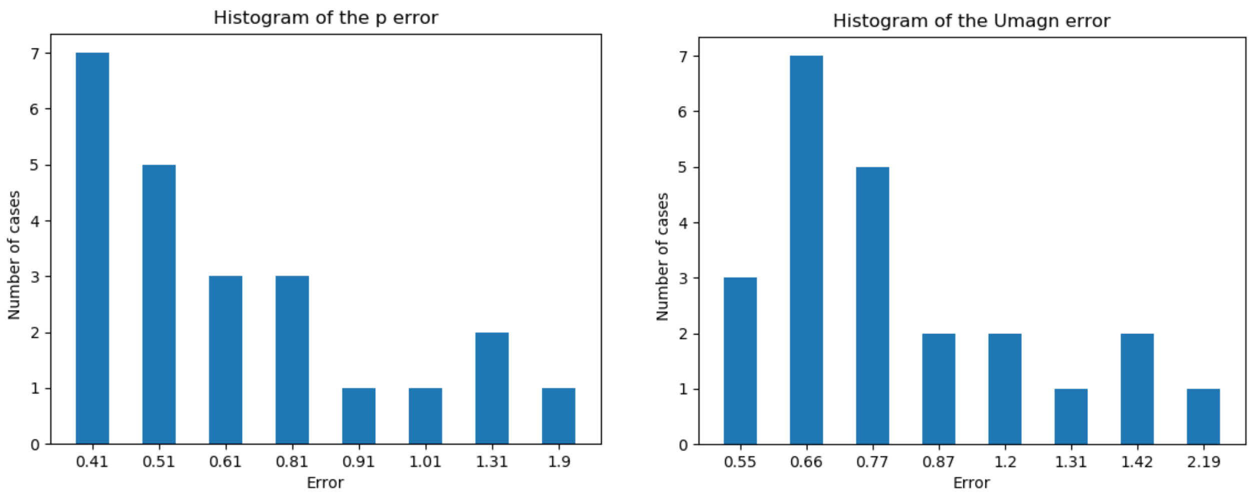


Figure 14: Histograms of the pressure (left) and the velocity magnitude (right) errors for the unknown cases. Due to the small size of the database, the number of the samples is denoted in the y axis instead of the percentage of the samples.

In Figure 15 contours of the pressure and the velocity magnitude fields, as well as their errors for sample 1070 are shown. Sample 1070 has been selected for visualization since it has a combination of new features compared to the train closed AGS data set that was utilized for training the model. The mean spatial error of sample 1070 is 0.44% for the pressure and 0.81% for the velocity magnitude. Overall, the prediction is satisfactory, although there are some isolated high error regimes. The geometry modifications of this sample, such as the front bumper x translation, the cowl x translation, and the underbody spare tire z translation drastically affect the flow field in the front bumper area, in the bonnet leading edge regime, and the flow field behind the battery respectively and thus, the spatial error in these regimes is increased.

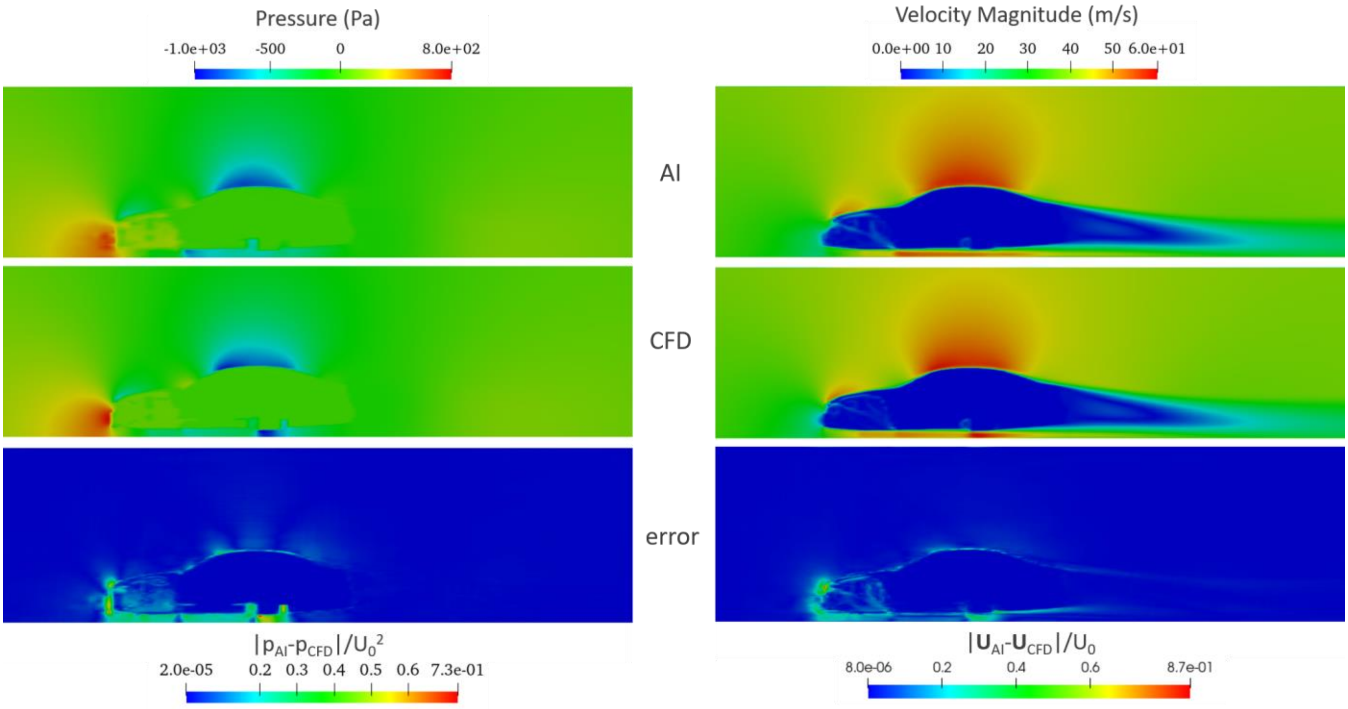


Figure 15: Pressure and velocity vector contour fields for case 1070 of the unknown database. On the left side, the predicted (AI, 1st line) and the target (CFD, 2nd line) pressure fields are demonstrated first and then the error is shown (error, 3rd line). Similarly, on the right side, the predicted (AI, 1st line) and the target (CFD, 2nd line) velocity magnitude fields are demonstrated first, and then the error is shown (error, 3rd line).

4. Higher fidelity application

4.1. Introduction and objectives

One of the goals in this stage of the project is to verify that the identified model order reduction method, that has been trained and verified on 2D models, is applicable to more relevant cases for industrial applications. The developed workflow to predict aerodynamic coefficients and flow fields based on machine learning approach is here applied to a “higher fidelity” model, and obtained results are compared with actual CFD results.

The vehicle chosen for this application is the 3D version of the electrified DrivAer model, more details on the model and its preparation can be found in D2.1: Requirements for aerothermal simulations reduced order model.

The object of the investigation is still a simplified model, compared to a full production vehicle, but it includes all the features of a realistic BEV car (heat exchangers, electric machines, battery pack, …); for this reason this test case can be considered valid for reduced order models validation in view of the final application in WP4.

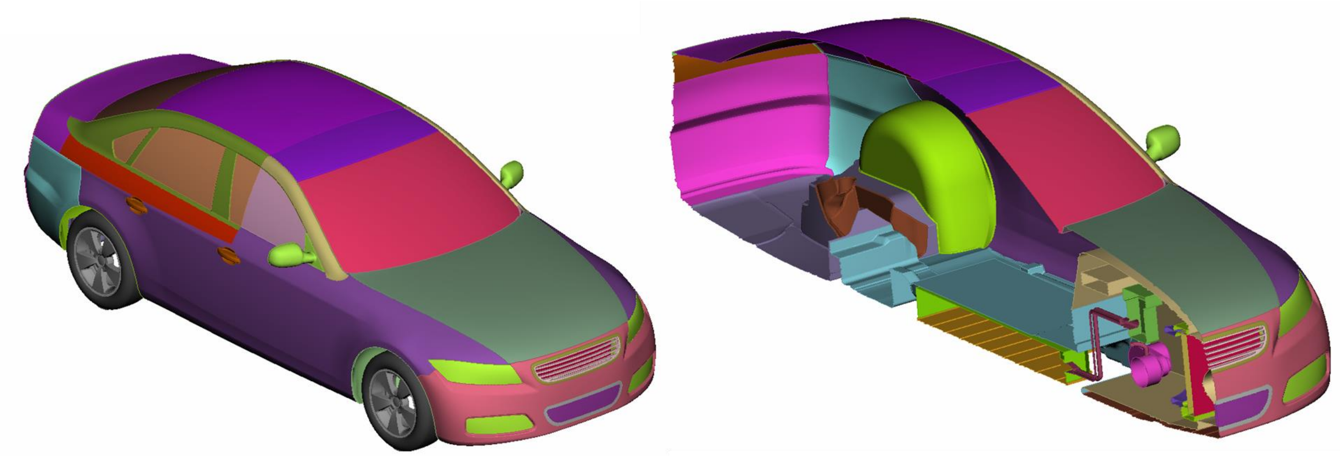


Figure 16: Overview of the external shape and underhood-underbody region of the electrified DrivAer model

4.2. Geometry dataset generation

A parametrization including thirteen different variables on the external surface is prepared on the model, according to the requirements deriving from WP1 and WP2 developed framework. At this stage of the project it has been chosen to proceed with the original parametrization of the model, not including yet the work under development in task 1.3 on the universal geometry parametrization. The reason of this choice is to guarantee the homogeneity of the evaluation of the method between 2D and 3D application, just extending in the third dimension the existing parameters.

An overview of the areas involved in the parametrization is shown in Figure 17. As visible, the modifications involve almost all the area in the symmetry line section that can significantly affect aerodynamics.

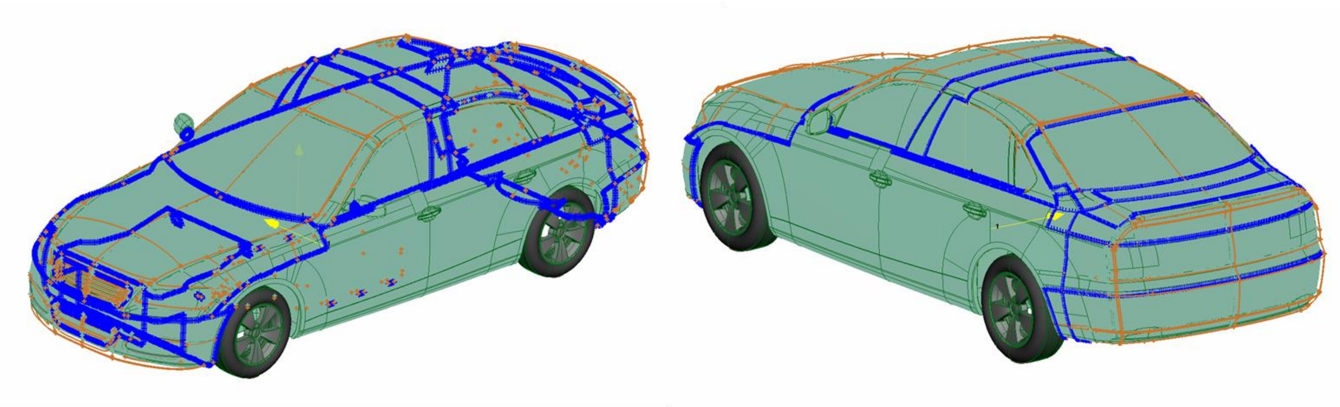
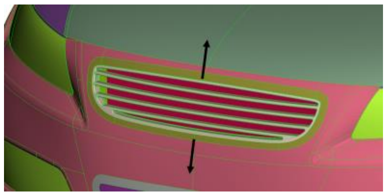
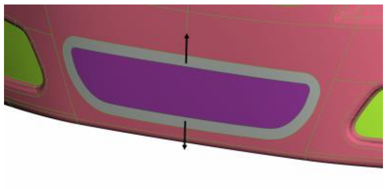
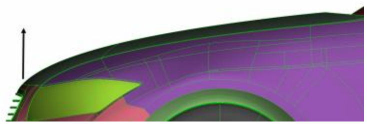
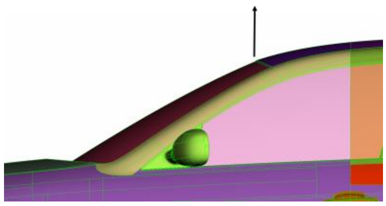
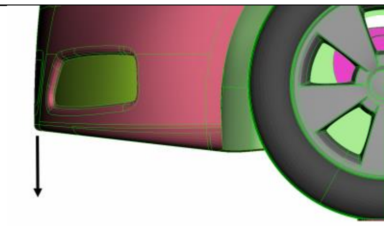
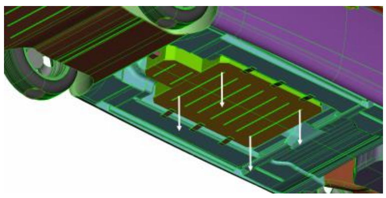
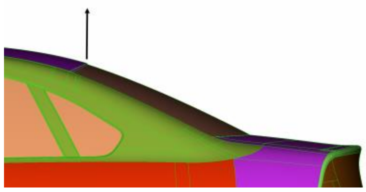
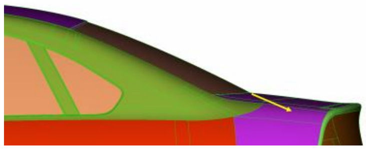
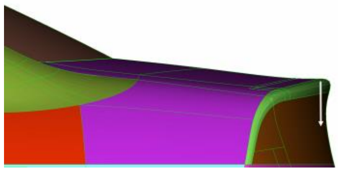
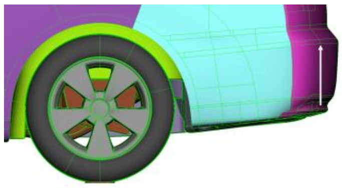
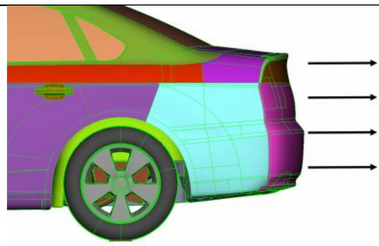
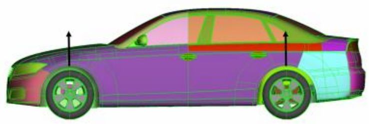
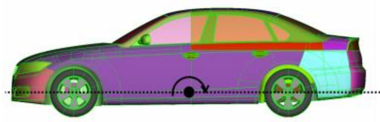


Figure 17: Global overview of the geometric parametrization on electrified DrivAer 3D model

Also, the possible range of modification of the parameters is chosen accordingly to what prescribed for the 2D model, for the above-mentioned reasons. Some random shapes are generated in advance to the full DoE in order to check that the morphing tool is able to produce good quality surfaces. The full list of parameters and ranges for this application is visible in Table 5.

Table 5: List of parameters and ranges on electrified 3D DrivAer model

ID	Parameter	Type		MinValue	MaxValue	Affected Area
1	PARAM-upperGrille-Z	DFM	EDGE-FIT	-14 mm (each side)	+10 mm (each side)	
2	PARAM-lowerGrille-Z	DFM	EDGE-FIT	-15 mm (each side)	+15 mm (each side)	
3	PARAM-bonnetLE-Z	DFM	EDGE-FIT	-32 mm	+48 mm	
4	PARAM-windscreen-angle	DFM	EDGE-FIT	-25 mm	+30 mm	
5	PARAM-ramp-angle	DFM	EDGE-FIT	-50 mm	+60 mm	
6	PARAM-batteryPack-Z	DFM	EDGE-FIT	-10 mm	+30 mm	
7	PARAM-rearWindowAngle	DFM	EDGE-FIT	-25 mm	+25 mm	

8	PARAM-rearWindowLength	DFM	EDGE-FIT	-45 mm	+45 mm	
9	PARAM-trunkLidAngle	DFM	EDGE-FIT	-30 mm	+48 mm	
10	PARAM-diffuserAngle	DFM	EDGE-FIT	-50 mm	+50 mm	
11	PARAM-trunkLength	DFM	EDGE-FIT	-50 mm	+50 mm	
12	PARAM-rideHeight-Z	DFM	TRANSLATE	-25 mm	+50 mm	
13	PARAM-rideHeight-angle	DFM	ROTATE	-8.0 mm (each side)	8.0 mm (each side)	

An example of application of the implemented morphing parameters is provided in the following images (Figure 18, Figure 19 and Figure 20): three different shapes included in the generated DoE are shown and compared with the baseline shape.

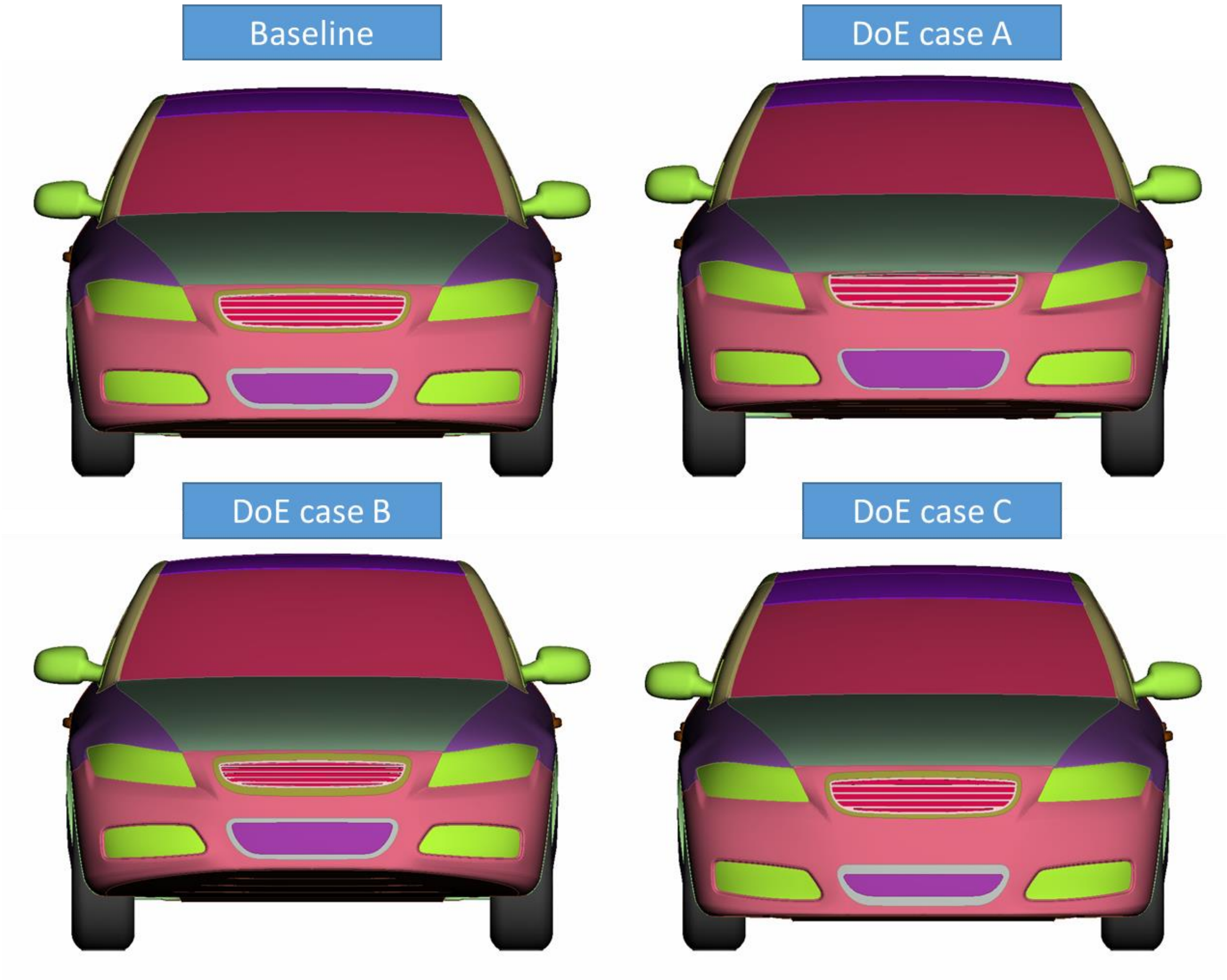


Figure 18: Comparison between baseline and DoE shapes on 3D DrivAer, front view

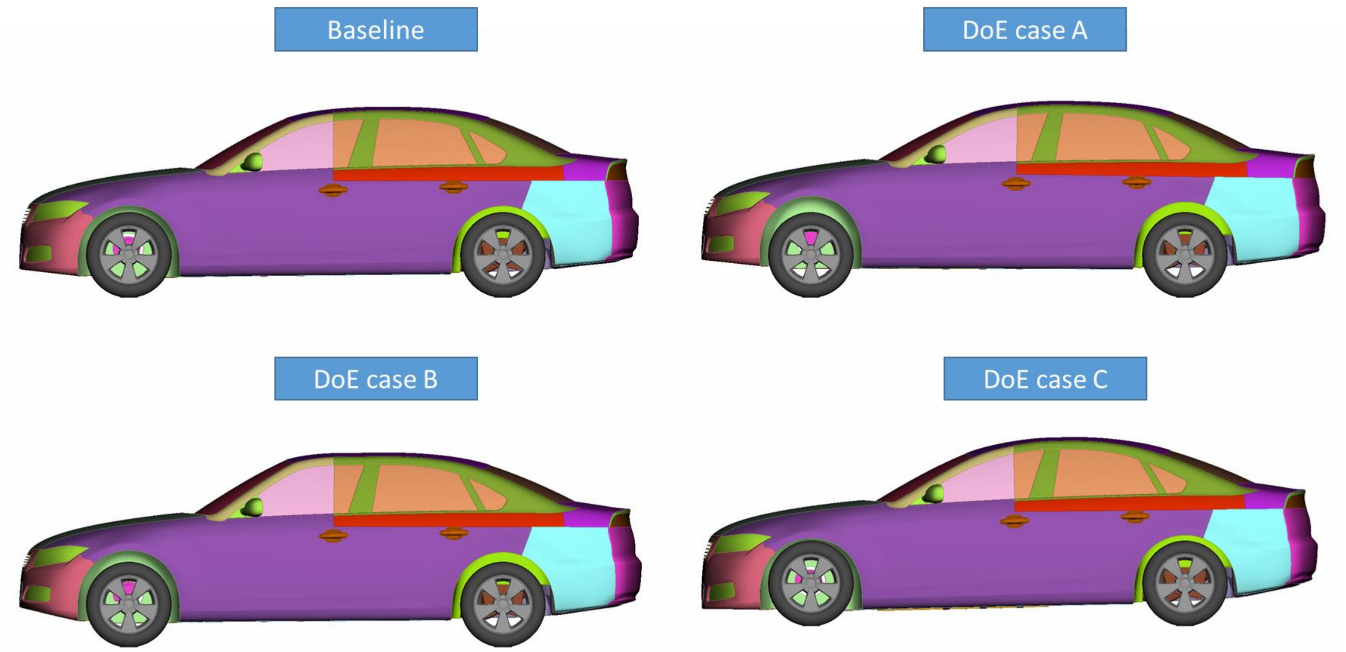


Figure 19: Comparison between baseline and DoE shapes on 3D DrivAer, side view

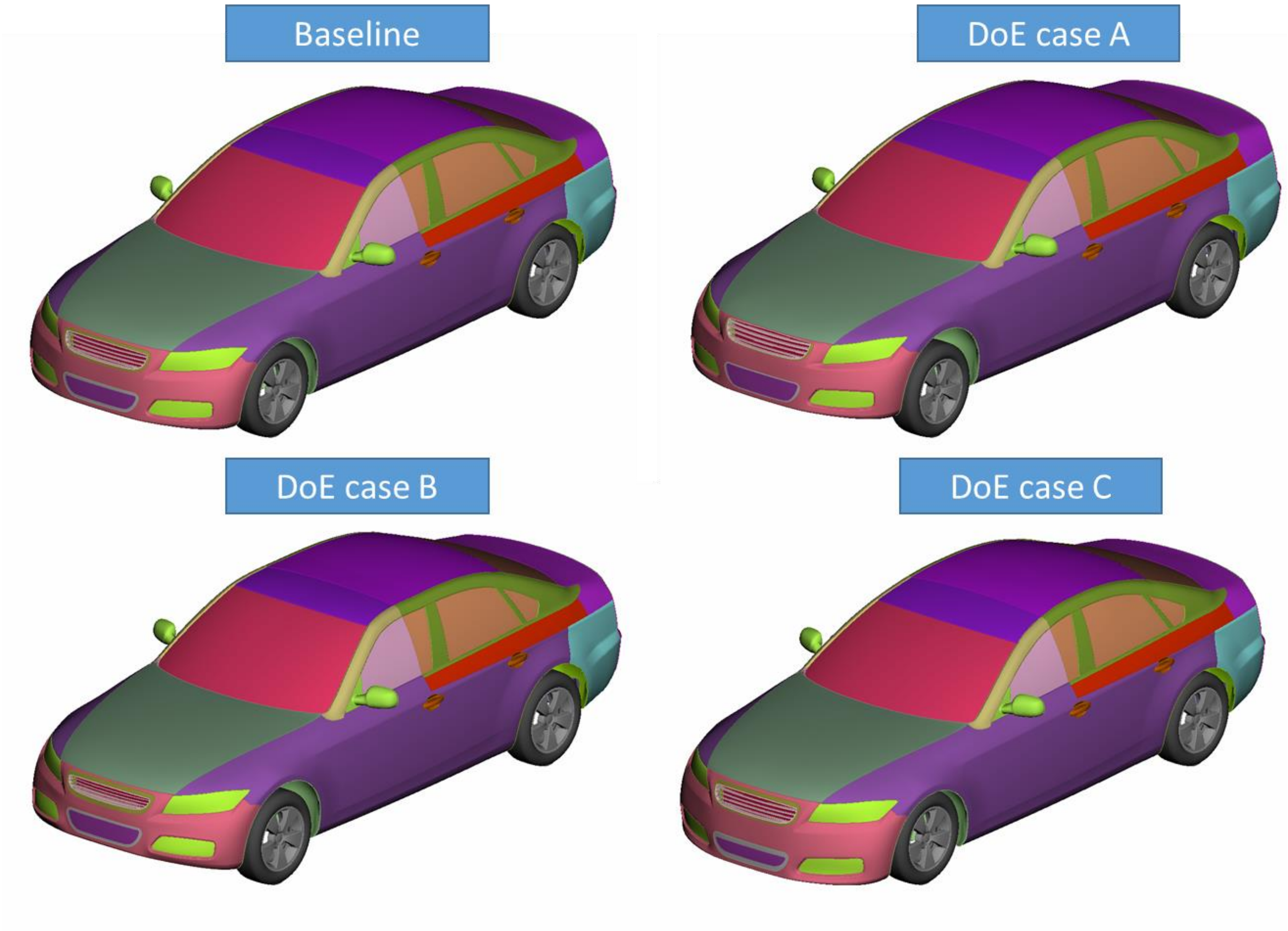


Figure 20: Comparison between baseline and DoE shapes on 3D DrivAer, isometric view

The design of experiments is planned accordingly to the 2D dataset workflow: a SOBOL sequence of one thousand shapes is generated, plus fifty additional spare points. For each configuration, the steady CFD workflow is applied a described in section 4.3.

4.3. CFD dataset generation

The computational cost in the higher fidelity application has been significantly increased, not only for the 3-D ANN training but also for the volumetric database generation, and therefore, the latter necessitates an accelerated platform for the vehicle CFD simulations. The proposed workflow, as can be seen in Figure 21, includes the development and enhancement of pre-processing utilities, the CFD solver, and data-extraction tools.
In the pre-processing context of the workflow, the computational grid for each vehicle geometry was constructed by employing the HELYX mesh generator (helyxHexMesh) [2], which is an evolved version of snappyHexMesh with improved performance and mesh quality. Since all geometry variants to be used in the study were derived from a single baseline case and their flow fields are expected to be similar, the mapping of initial fields from previous results was adopted instead of potential flow for field initialization. The newly developed mapping technology (helyxMap) uses a K-nearest neighbour (KNN) search [3] and it is up to 10 times more efficient than traditional Octree mapping techniques (mapFields), as it has been indicated by benchmark tests in a 100M case. The way the mapping is performed in the new utility is either coordinate

or wall distanced based. In the coordinate-based map utility, areas of similar location are mapped from the source to the target, whereas in the wall-distance-based approach, areas of similar wall distances are mapped. The latter produces a smoother initial solution as it avoids mapping a solid region into a fluid one or vice versa.
Regarding the CFD solver selected for creating the flow field DOEs, a pressure-based block solver [4], [5] with Algebraic Multi-Grid was employed. In the 3-D steady-state vehicle simulations that were performed as part of this work, the coupled solver was found to converge twice as fast as compared to the heavily optimized segregated counterpart. Due to faster convergence and incremental speed-up per case, the total simulation time was significantly reduced. Furthermore, a new convergence assessment criterion, based on statistical and integral considerations, was applied to terminate the simulation once objective conditions regarding the aerodynamic coefficients were satisfied, so avoiding redundant iterations at the end of the calculation.
As a final step, the CFD solution on the cell centres and the vehicle geometry was sampled in the vicinity of the vehicle, significantly reducing the volume of data that has to be stored for subsequent use with the AI training algorithms. The sampling can be either structured by employing the KNN algorithm (helyxSample) or unstructured by making use of OpenVDB technology [6]. In both cases, the computational cost compared to traditional sampling techniques used in OpenFOAM is significantly reduced.
The efficiency of the workflow is further assessed in the accelerated solver deliverable (D1.2).

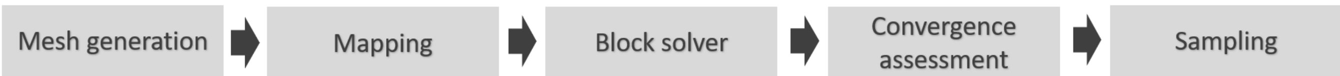


Figure 21: Workflow of the 3-D DOE: the utilities and the CFD solver.

Due to memory limitations in the GPU accelerators and the fact that the currently developed ML algorithms cannot run in parallel in multiple GPU accelerators or even in multiple GPU nodes, a relatively low-resolution input has been used for the training of the ANN. If a higher resolution database was used, it would allow simultaneous training of only a few multiple samples and as a result of the reduced batch size, the accuracy of the gradient computation during backpropagation would be dramatically affected. Apart from that, the model size and the stored parameters alongside the gradients would have outstandingly risen. Developments to increase the batch size by accumulating the gradients of several mini-batches, and secondly, by parallelizing the ML algorithm for multiple GPUs are in progress, but they are out of the scope of the current report. Furthermore, efforts to parallelize the model are also being made.

Based on the above, two databases with different resolutions have been utilized. In the lower resolution database (drivAer_3D_low), uniform sampling (64x64x64 points) in a box containing the vehicle is applied, whereas in the higher resolution database (drivAer_3D_high), uniform sampling (128x64x256 points) in a box containing half of the vehicle is performed, taking advantage of plane symmetry at y=0. After memory profiling during the training process for the higher resolution database, it was found that most of the GPU memory was occupied and the batch size was reduced to make the training feasible.

4.4. Drag coefficient prediction

The results of the training for predicting the drag coefficient in the drivAer 3-D database are presented in this section. The training settings are the same as in section **¡Error! No se encuentra el origen de la referencia.**. In Table 6 the mean error and its standard deviation, as well as the training wall clock time for the low and high-resolution 3-D drivAer databases are shown. It can be concluded that by doubling the sampling points in each dimension, the computational cost is increased by a factor of 9.5 and the accuracy in the drag prediction is improved by approximately 35%. In addition, the standard deviation of the error remains smaller than the mean error value in both cases. As it can be elucidated from Figure 22, where the histograms of the drag coefficient error are shown for the 2 databases, the maximum error is also significantly reduced from 13.2% to 4.9%. Overall, the results are satisfactory as the drag coefficient predictions are 98.4% and 98.8% accurate for the low and the high-resolution databases respectively.

Table 6: Effect of the database resolution on the drag prediction using 5 convolution layers and the adaGrad optimizer. The error, the standard deviation of the error, as well as the training wall clock time are shown.

Database	C_D (error %)	std($C_{D,error}$)	Wall clock time (min)
drivAer_3D_low	1.57	1.45	77.12
drivAer_3D_high	1.15	0.94	731.22

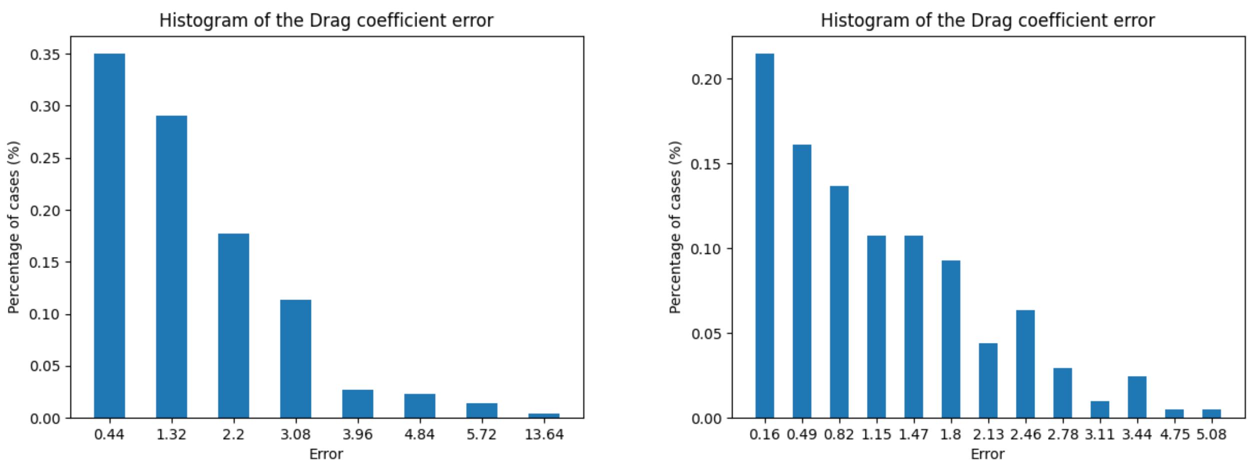


Figure 22: Histograms of the Drag coefficient error for the low (left) and the high (right) resolution data sets.

4.5. Flow fields prediction

The architecture of the ANN and the training configuration as described in the Flow prediction algorithm part of section 3.2.1 in deliverable D2.2, is followed in general here as well. The different NN parameters used for training the 3-D models are shown in Table 7. Due to the increased model and input, outputs sizes, the batch size for the finer resolution training is reduced, otherwise, the memory requirements for the training would exceed the available GPU memory and the training would be terminated.

Regarding the last up-sampling layer, for the transposed convolution operator, $K_t=2, S_t=2, P_t=0, D_t=1$ instead of $K_t=4, S_t=2, P_t=1, D_t=1$ were used in order to reduce the checkerboard artifacts. These artifacts are caused due to uneven overlapping during the up-sampling process. The phenomenon of the checkerboard artifacts is enhanced in 3 dimensions as the uneven overlaps of each dimension are multiplied together [7]. It can be elucidated from Figure 24, where the error contour fields for the two different settings are shown, that the artifacts are significantly diminished when using the new transpose convolution configuration. The reduced artifacts result in increased accuracy of the predicted flow fields and smaller mean absolute error loss as can be seen in Table 8 and Figure 23 respectively. Furthermore, from Figure 23 it can be inferred that the error loss in the validation set is converging to the error loss of the training set when the new transpose convolution configuration is applied, which is not the case when the old settings are kept.

Next, the new configuration for the transpose convolution is also applied for training the finer resolution database. In Table 8, the errors and their standard deviation for the flow fields, as well as the training wall clock times are compared for the two databases. It can be deduced that by doubling the sampling points in each dimension, the pressure field error is reduced by 11%, the velocity vector field error decreases by 26%, while the computational time is increased by 9 times. The mean absolute error loss of Figure 25 for the high-resolution database follows the same trend as the low-resolution database in Figure 23, since in both cases the validation loss converges to the training loss. Some additional findings for the high-resolution training follow: the histograms for the errors of the pressure and the velocity vector fields are shown in Figure 26, whereas in Figure 27 and Figure 28, the flow fields contours on two different slices for sample 730 are presented. Sample 730 was selected for visualization as it was found to be the case with the worst predictions for both pressure and velocity magnitude fields, and the corresponding errors were found to be 0.154% and 0.658%. In accordance to the 2-D findings, discrepancy between the target (CFD) and the predicted (AI) fields was noticed along the surface of the car. In addition, areas of high error were identified around the side mirror and in the engine bay for both the pressure and the velocity fields, while high velocity errors were also noticed in the shear layer leaving the trunk.

Table 7: NN parameters used for the flow prediction training for the 2 different databases.

NN parameters	drivAer_3D_low	drivAer_3D_high
batch_size	*16*	*10*
levels	*5*	*6*
learningRate	*0.00002*	*0.00002*

Table 8: Effect of the different transpose convolution settings on the accuracy of the flow prediction variables. The errors in the pressure and velocity magnitude prediction are shown. The low-resolution database has been employed.

transConv3d	**p** (error %)	**U** (error %)
(4, 2, 1)	0.450	1.170
(2, 2, 0)	0.142	0.570

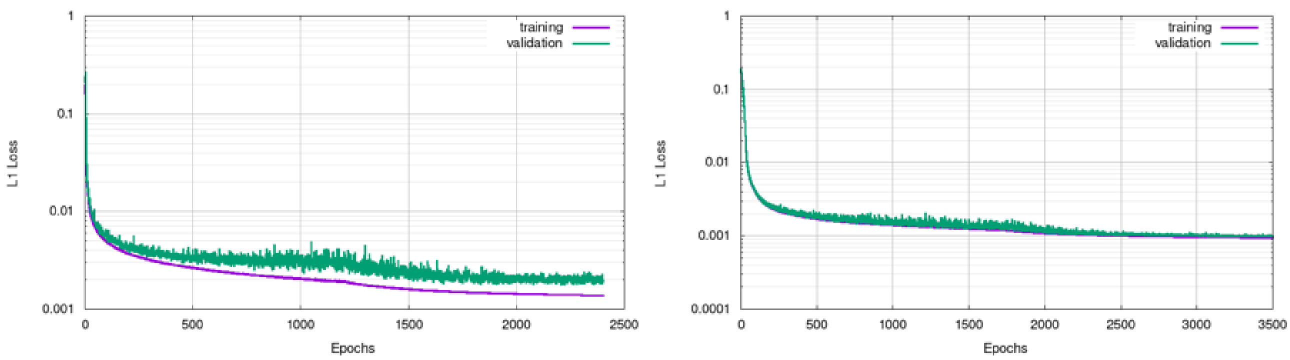


Figure 23: L_1 loss for the two permutations in the last up-sampling layer (low-resolution database). On the left figure *K_t=4, S_t=2, P_t=1, D_t=1 setting were used, whereas on the right figure K_t=2, S_t=2, P_t=0, D_t=1 were used during transpose convolution*.

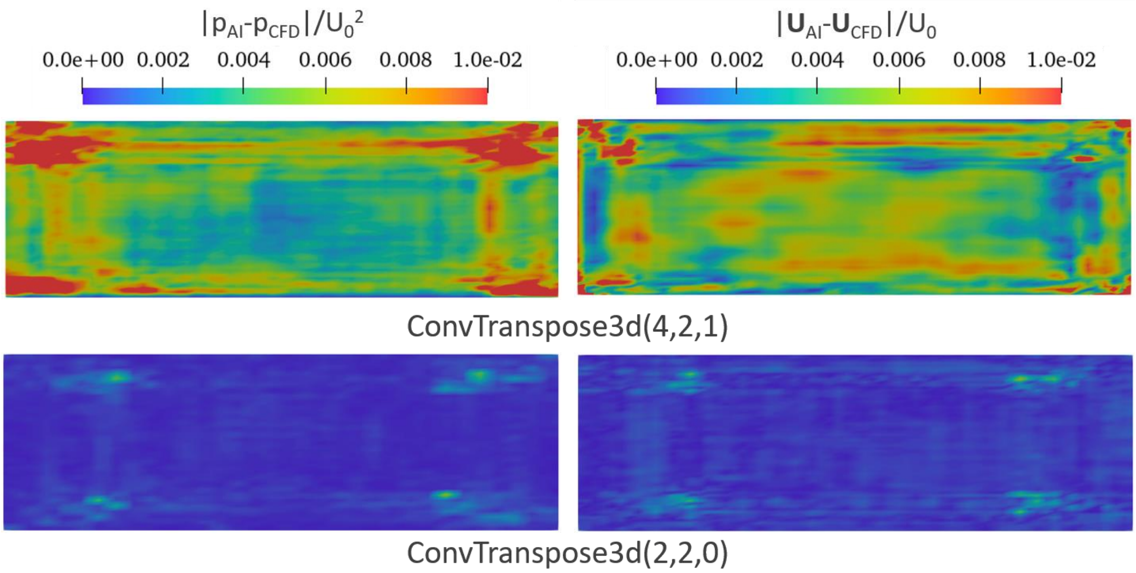


Figure 24: Contour fields of the pressure (left) and velocity (right) errors at z=3.27 plane for case 101 using the low-resolution database. Training with transpose convolution settings: *K_t=4, S_t=2, P_t=1, D_t=1 were used in the first row, whereas in the second row the transpose convolution setting used* are: *K_t=2, S_t=2, P_t=0, D_t=1*.

Table 9: Effect of the different data sets on the accuracy of the flow prediction variables. Error in the pressure and velocity magnitude prediction, the standard deviation of these errors, as well as the training wall clock time are shown in the corresponding columns.

Database	**p** (error %)	**U** (error %)	**std(p_{error})**	**std(U_{error})**	**Wall clock time (min)**
drivAer_3D_low	0.142	0.570	0.007	0.041	887
drivAer_3D_high	0.126	0.417	0.007	0.048	8199

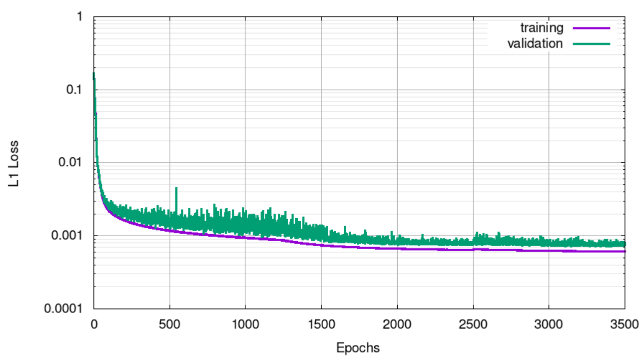


Figure 25: L_1 loss for the new transpose convolution configuration (high-resolution database).

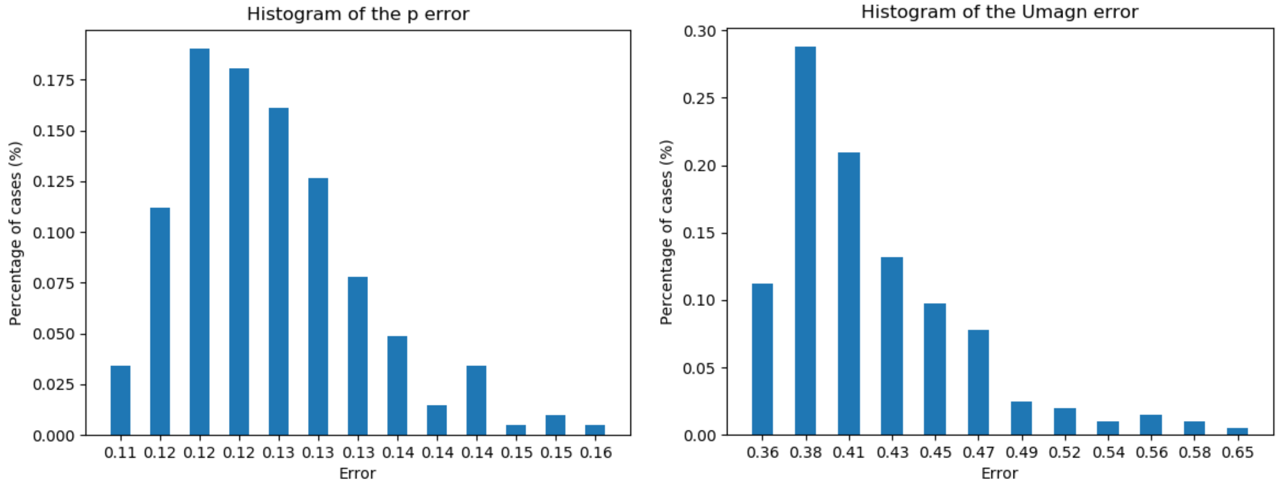


Figure 26: Histograms of the pressure (left) and the velocity magnitude (right) errors for the high-resolution 3-D database.

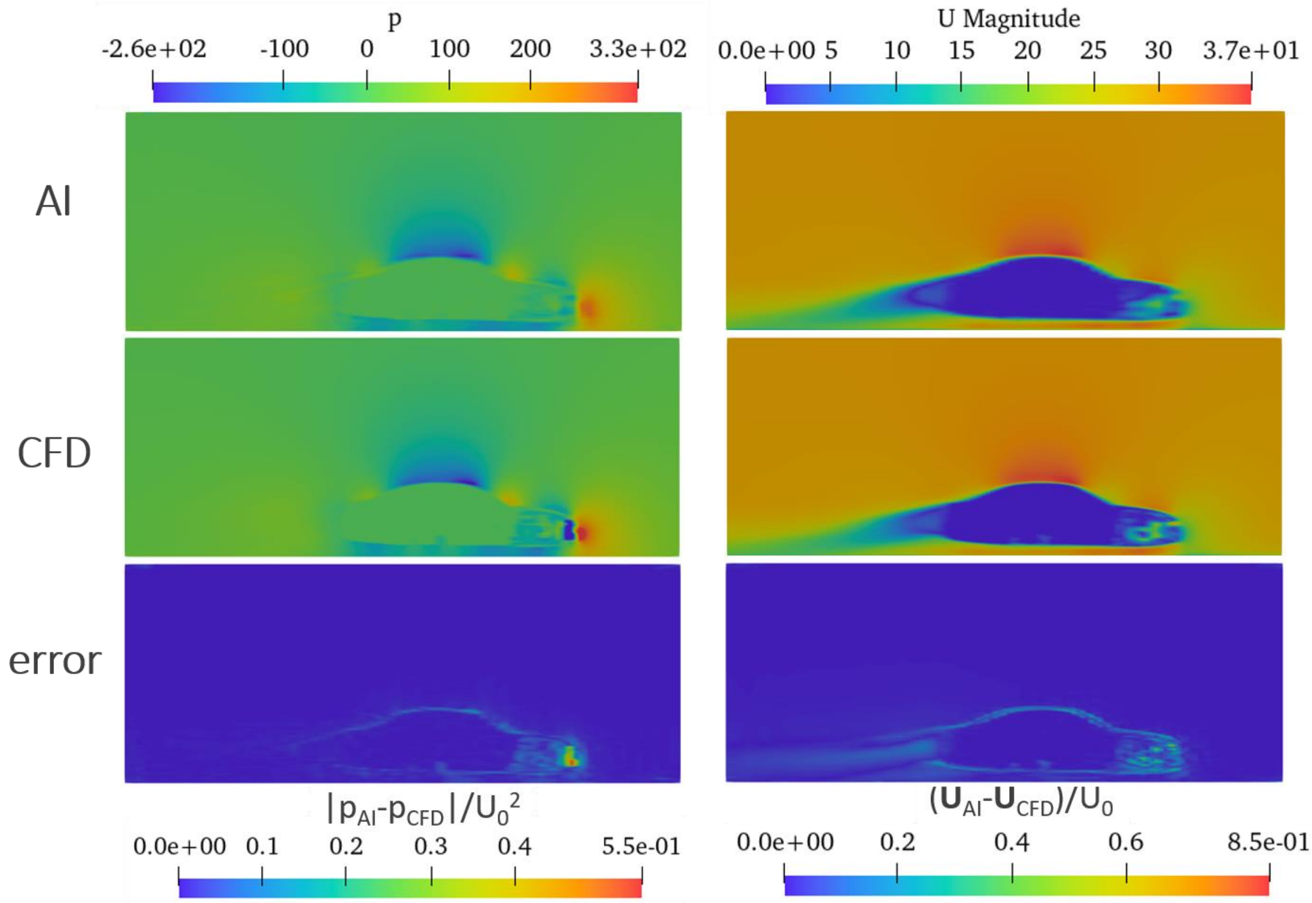


Figure 27: Pressure and velocity vector contour fields on plane y=0 for case 730 of the high-resolution database. On the left side, the predicted (AI, 1st line) and the target (CFD, 2nd line) pressure fields are demonstrated first and then the error is shown (error, 3rd line). Similarly, on the right side, the predicted (AI, 1st line) and the target (CFD, 2nd line) velocity magnitude fields are demonstrated first, and then the error is shown (error, 3rd line).

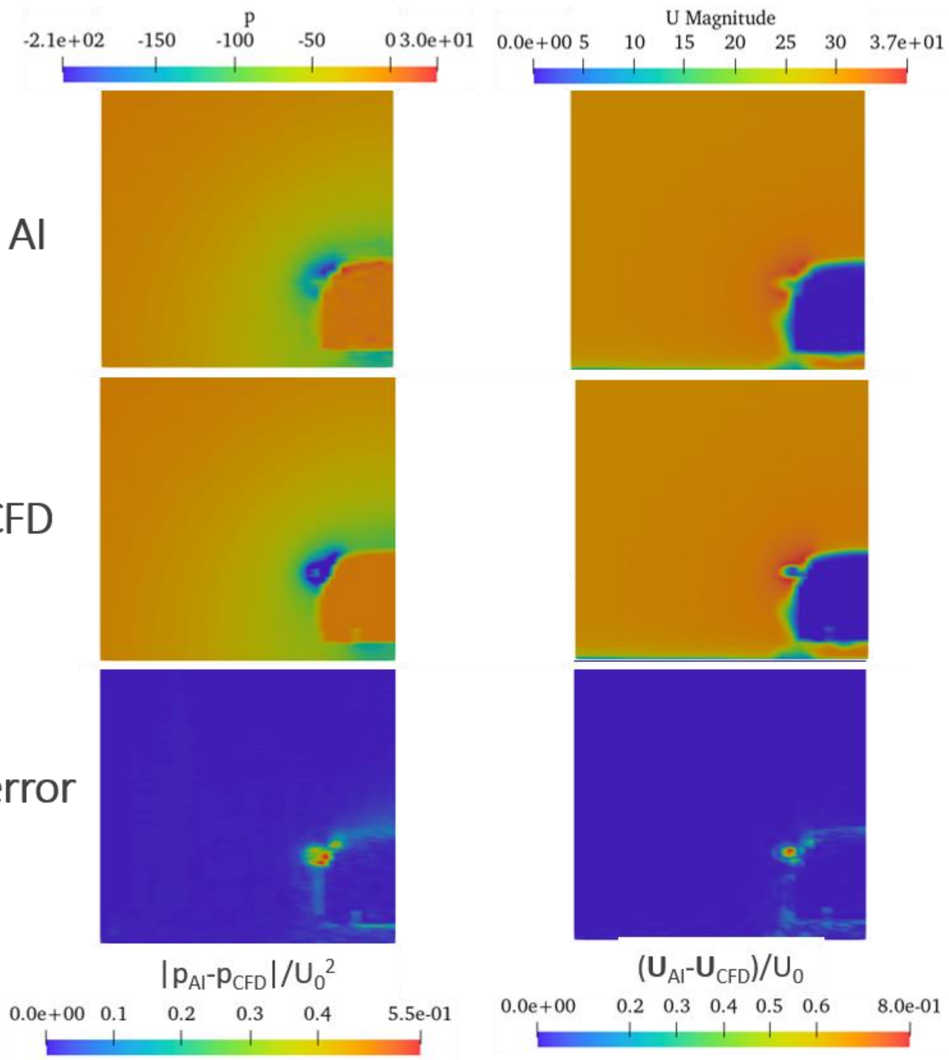


Figure 28: Pressure and velocity vector contour fields on plane x=0.99 for case 730 of the high-resolution database. On the left side, the predicted (AI, 1st line) and the target (CFD, 2nd line) pressure fields are demonstrated first and then the error is shown (error, 3rd line). Similarly, on the right side, the predicted (AI, 1st line) and the target (CFD, 2nd line) velocity magnitude fields are demonstrated first, and then the error is shown (error, 3rd line).

4.6. Final assessment on implemented workflow

In conclusion, the benchmark tests on the 2-D geometries demonstrated the capability of both methods to accurately predict the drag coefficient and the volumetric flow fields. Between the two methods, which denoted similar accuracy in the flow field predictions, NN approach was chosen as the most appropriate for performing the 3-D predictions, despite its computational cost. The ML algorithms have demonstrated the capability to accurately predict the drag coefficient and the volumetric flow fields for samples out of the design space, which justifies their selection over the POD+I method. In an effort to prove the applicability of the NN approach to real vehicle predictions of WP4, high fidelity simulations for the drivAer 3-D geometry have been also performed as a preliminary step. The latter required an optimized DOE platform for CFD simulations to significantly reduce the computational time for generating the database. The 3-D models which followed, demonstrated consistent results with the 2-D cases studied and evidence that the current methodology can be delivered to WP4.

5. Conclusions and future works

The 2.3 task bridges the gap between simplified tests undertaken by means of 2D modelling and the application on real vehicle designs undertaken in WP4. 2D models were used to check the feasibility of ML tools to predict the aerodynamic forces and flow-fields at proof of concept level and then the research moved to 3D models in order to check the limitations of a ML framework regarding computational costs and accuracy. Jumping to 3D applications required efficient processes regarding both, the generation of data by means of CFD simulation and the ML training process. The accelerated CFD solver has been proved successful and has allowed the simulation of hundreds of complete vehicle simulations in reasonable times. The challenge of the ML algorithm of managing one order of magnitude higher memory usage of 3D models compared with 2D models has also been successfully demonstrated. Finally, the goal of achieving error levels below 2% has been achieved not only for the aerodynamic forces but for the velocity and pressure field predictions as well. After confirming the feasibility of using a ML-based aerodynamic ROM with 3D models, this framework will be applied to 3 real designs provided by the consortium OEMs in WP4, including a sedan, a city car and a SUV.

ACKNOWLEDGEMENT

The author(s) would like to thank the partners in the project for their input, valuable comments on previous drafts and for performing the review.

Project partners:

PARTICIPANT Nº	PARTICIPANT ORGANISATION NAME	COUNTRY
1 (Coordinator)	IDIADA AUTOMOTIVE TECHNOLOGY SA (IDIADA),	Spain
2	VOLVO PERSONVAGNAR AB (Volvo Cars)	Sweden
3	VOLKSWAGEN AG (VW)	Germany
4	CENTRO RICERCHE FIAT SCPA (CRF)	Italy
5	ESI GROUP (ESI GROUP)	France
6	ENGYS LTD (ENGYS LTD)	United Kingdom
7	Kompetenzzentrum - Das Virtuelle Fahrzeug, Forschungsgesellschaft mbH (VIF)	Austria
8	VRIJE UNIVERSITEIT BRUSSEL (VUB)	Belgium
9	ECOLE NATIONALE SUPERIEURE D'ARTS ET METIERS (ENSAM PARISTECH)	France
10	ALGORITHMICA TECHNOLOGIES GMBH (ALGORITHMICA)	Germany
11	F INICIATIVAS I MAS D MAS I SL (F-INICIATIVAS)	Spain

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement no. 824306

6. References

1. *Introduction of a New Realistic Generic Car Model for Aerodynamic Investigations.* A. Heft, T. Indinger, N. Adams. Detroit, Michigan, USA. s.n., April 23-26, 2012.
2. Automatic Parallel Polyhedral Mesh Generation on Complex Geometries in OpenFOAM. E. de Villiers, A. Jackson and M. Janssens. Third OpenFOAM Workshop, Politecnico di Milano, 2008.
3. *Approximate Nearest Neighbor Searching*. S. Arya and D. M. Mount. Proc. 4th Annu. ACM-SIAM Sympos. on Discrete Algorithms (SODA'93), 1993, 271-280.
4. *A Hybrid Method for Shape And Topology Optimization in Fluid Mechanics*. G.K.Karpouzas. Ph.D. Thesis, Engys Ltd. - NTUA, 2018.
5. *A coupled finite volume solver for the solution of incompressible flows on unstructured grids*. M. Darwish, I. Sraj, and F. Moukalled. Journal of Computational Physics, 2009.
6. *VDB: High-Resolution Sparse Volumes with Dynamic Topology*. K. Museth. ACM Transactions on Graphics 32(3), 2013. Presented at SIGGRAPH 2013.
7. *Convolutional Neural Networks Without Any Checkerboard Artifacts. Y. Sugawara, S. Shiota and H. Kiya.* 2018 26th European Signal Processing Conference (EUSIPCO)*, Rome, Italy, 2018, pp. 1317-1321, doi: 10.23919/EUSIPCO.2018.8553099.*