



HAL
open science

Enhanced Drag Force Estimation in Automotive Design: A Surrogate Model Leveraging Limited Full-Order Model Drag Data and Comprehensive Physical Field Integration

Kalinja Naffer-Chevassier, Florian de Vuyst, Yohann Goardou

► To cite this version:

Kalinja Naffer-Chevassier, Florian de Vuyst, Yohann Goardou. Enhanced Drag Force Estimation in Automotive Design: A Surrogate Model Leveraging Limited Full-Order Model Drag Data and Comprehensive Physical Field Integration. *Computation*, inPress, 12 (10), pp.207. 10.3390/computation12100207. hal-04708986

HAL Id: hal-04708986

<https://utc.hal.science/hal-04708986v1>

Submitted on 25 Sep 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0
International License

Enhanced Drag Force Estimation in Automotive Design: A Surrogate Model Leveraging Limited Full-Order Model Drag Data and Comprehensive Physical Field Integration

Kalinja Naffer-Chevassier ^{1,2,†,‡}, Florian De Vuyst ^{3,‡}  and Yohann Goardou ^{1,*}

¹ Renault Group, Technocentre Renault, 78280 Guyancourt, France

² Université de Technologie de Compiègne, LMAC, CS 60319, 60203 Compiègne, France

³ Université de Technologie de Compiègne, CNRS, BMBI, CS 60319, 60203 Compiègne, France

* Correspondence: kalinja.naffer-chevassier@renault.com; fdevuyst@utc.fr; yohann.goardou@renault.com

† Current address: Affiliation 2.

‡ These authors contributed equally to this work.

Abstract: In this paper, a novel surrogate model for shape-parametrized vehicle drag force prediction is proposed. It is assumed that only a limited dataset of high-fidelity CFD results is available, typically less than ten high-fidelity CFD solutions for different shape samples. The idea is to take advantage not only of the drag coefficients, but also physical fields such as velocity, pressure and kinetic energy evaluated on a cutting plane in the wake of the vehicle and perpendicular to the road. This additional ‘augmented’ information provides a more accurate and robust prediction of the drag force, compared to a standard surface response methodology. As a first step, an original reparametrization of the shape based on combination coefficients of shape principal components is proposed, leading to a low-dimensional representation of the shape space. The second step consists in determining principal components of the *x*-direction momentum flux through a cutting plane behind the car. The final step is to find the mapping between the reduced shape description and the momentum flux formula to achieve an accurate drag estimation. The resulting surrogate model is a space-parameter separated representation with shape principal component coefficients and spatial modes dedicated to drag-force evaluation. The algorithm can deal with shapes of variable mesh, by using an optimal transport procedure that interpolates the fields on a shared reference mesh. The Machine Learning algorithm is challenged on a car concept with a shape design space of dimensional three. With only two well-chosen samples, the numerical algorithm is able to return a drag surrogate model with reasonable uniform error over the validation dataset. An incremental learning approach involving additional high-fidelity computations is also proposed. The leading algorithm is shown to improve the model accuracy. The study also shows the sensitivity of the results with respect to the initial experimental design. As a feedback, we discuss and suggest what appear to be the correct choices of experimental designs for best results.

Citation: Lastname, F.; Lastname, F.; Lastname, F. Title. *Computation* **2024**, *1*, 0. <https://doi.org/>

Keywords: Automotive engineering; drag force; surrogate model; reduced-order model; machine learning; data-driven; limited data; shape parameters; far field; optimal transport

Received:

Revised:

Accepted:

Published:

Copyright: © 2024 by the authors. Submitted to *Computation* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In automotive Engineering, drag reduction is a crucial step aimed at improving aerodynamic to increase a vehicle range. As it has a major impact on architecture and design of the product, aerodynamic is among the first performances to be studied and optimized. The design convergence is split between physical and numerical testing. While physical testing allows for faster iteration and more precise results, it is limited by the possibilities offered in a wind tunnel facility. It is often used in final stages of the development, with fine-tuning and approval in mind. On the contrary, numerical simulation is used very early in a project to initiate the design convergence, by eliminating non-compliant design and

propose solutions to improve the aerodynamic performance. However each simulation is time-consuming and costly. A typical automotive simulation involves turbulence models and unsteady computation, leading to restitution delay in hours. Shape optimization is often used in this context to propose small evolutions on various locations to improve the global performance. The design variables are mostly defined by the three-dimensional CAD shape parametrization. For standard vehicles, the admissible design domain can be rather big with numerous parameters. Among the reference computational approaches for shape optimization, one can distinguish two main families of methods: gradient-based methods and surrogate modeling techniques.

1.1. Gradient-based approaches

The first family considers gradient-based or descent-based optimization algorithms. These iterative methods require at least the evaluation of the gradient of the cost function (the drag coefficient in the context of this paper) with respect to the shape variables. For the evaluation of the gradient, adjoint-based methods have marked a significant progress in optimization techniques. The calculation of the gradient using the adjoint method was introduced by Lions [1], then further developed e.g. by Pironneau [2] and Jameson [3], [4]. The key feature of the adjoint method is that the numerical complexity to evaluate the gradient does not depend on the number of variables, allowing its application to high-dimensional parametric problems. Although extensively utilized in the aeronautical field, this method is still relatively underused in automotive aerodynamics. The adjoint method requires the development of an additional code, the so-called adjoint code. The case of adjoint methods for unsteady computations is a harder task since the primal time-dependent solution has to be stored. Let us mention in this context the work by Cheylan et al. [5] where an adjoint solver has been derived from a Lattice Boltzmann Method code for the Navier-Stokes equations, including a large eddy simulation (LES) turbulence model. However, adjoint-based shape optimization is limited in practical applications due to the iterative and computationally expensive nature of CFD simulations. This makes it difficult to meet the demands of fast, interactive design optimization, where rapid feedback is essential. Moreover, in the automotive industry, design is often driven by aesthetic constraints, which are difficult to quantify mathematically. This makes it challenging to integrate them directly into an optimization process.

1.2. Surrogate modeling

The second family of methods groups together all the surrogate modeling techniques. A surrogate model is based on approximation methods such as regression to build an easy-to-compute approximate cost function. For that, a design of computer experiment (DoCE) is used. The first step is to choose a sampling strategy of the admissible design space. Then high-fidelity (HF) computations are run to evaluate the cost function at these sample points. The so-called Response Surface Methodology (RSM) is widely used in the Computer-Assisted Engineering community because of its simplicity of design and setup. RSM can be built with various approximation method such as Radial Basis Function [6], [7], Gaussian Process [8] or AI tools such as regression-based Artificial Neural Networks [12].

However, for high-dimensional design problems, a significant number of HF computations is needed. Latin Hypercube Sampling (LHS) strategy allows to sample a hypercube domain with a number of samples equal to the dimension d . However, it is known to be generally not sufficient to get an accurate response surface. An acknowledged empirical law is to consider a number of samples equal to d times a constant (between 3 and 5 for aerodynamic studies). In the automotive industry, this leads to an unrealistic budget of computations. This is why innovative methods that require fewer input samples, such as the approach proposed in this paper, are being explored. By approximating the results of expensive and time-consuming CFD simulations, surrogate models enable the creation of rapid, interactive design tools that significantly speed up the evaluation and optimization process.

1.3. Shape parametrization

Another important topic is the way to parametrize the car shapes. The performance of the optimization process can strongly depend on the shape parametrization. Some CAD parameters variations may have a very weak influence on the drag value. Of course the "true" manifold of parameters of interest is not known a priori. Moreover, for technical and practical reasons, engineers may only consider and manipulate meshed geometries as 'objects-of-study'. For different CAD samples, the mesh software may produce meshes with different mesh connectivity. Sometimes, the CAD parameters change during the study. If the surrogate model is built with those initial parameters, any changes would necessitate retraining the model. In that case, one has to imagine a reparametrization of the vehicle shapes from the available meshes. This will be discussed in this paper.

1.4. Adding information from available volume fields

As mentioned above, surface response methodologies can build regression functions of drag coefficients from various shape parameter sample HF computations. By solely retrieving aerodynamic coefficients from high-fidelity calculations, a significant part of the information contained in the results is lost. It appears advantageous to leverage ancillary information such as velocity fields, pressure distributions, or kinetic energy. As drag coefficients are correlated with these fields, one can expect more accurate drag force predictions providing correlated information. Volume fields defined on meshes are highly dimensional by nature. In order to get a low-dimensional representation of them, dimensionality reduction techniques such as principal component analysis (PCA, [9]), proper orthogonal decomposition (POD, [10]), locally linear embedding (LLE, [11]), multidimensional scaling or neural-based encoders [12] can be used. In the case of parametric problems, one can combine both dimensionality reduction with response surface methods to predict the reduced variables to get a parametric reduced-order model of the field.

As presented later on in Section 2 (methodology), the whole drag force is the integral of the normal stresses exerted onto the vehicle skin. Thus the drag force is a function of the pressure field and the velocity gradient. A way to derive a more physical drag force is to achieve a low-order surrogate model of the skin normal stresses. Another way to evaluate the drag force is to adopt a so-called *far field* approach where the drag force can be well-approximated by the integral of a difference of momentum fluxes over an arbitrary cutting plane, preferably located in the wake of the vehicle (this will be detailed and explained in Section 2). In this case, the drag force depends on the pressure distribution, velocity and kinetic energy fields. A key advantage of far field formulas is that the cutting plane can be the same for all the car shapes of interest, allowing us for an easier principal component analysis and dimensionality reduction.

1.5. Scope, objectives and structure of the paper

Classical RSM for drag prediction often require extensive input data, with hundreds to thousands of evaluations (simulations) needed to achieve acceptable accuracy. This evaluation-intensive requirement is not compatible with industrial processes where development times are short. Generating large datasets is not only time-consuming but also financially expensive. In the literature, there is a lack of surrogate models trained with a limited amount of input data, ideally of the order of the number of shape parameters. A big shortcoming of RSM is that they usually only make use of quantities of interest (here drag forces) that are often scalar, with all the physical fields being lost. Moreover, many current surrogate models are designed for predicting drag of cars with *significant geometric modifications* or even different vehicle models. These models are useful in the early stages of development to assess the bulk geometry. However, during the fine styling phase, where modifications are minimal (often just a few centimeters) these models become less effective. There is a clear gap in the literature regarding models that can accurately predict drag for such small geometric changes and with only limited input data. To overcome this scientific challenge, we have to parameterize the car shape in such a way that drag force

responses are strongly correlated to the shape parameters. There is also a constraint of low-dimensional representation since the dataset is made of few evaluations and one can only consider few degrees of freedom for the identification of the operator that maps shape into drag force.

These constraints lead us to consider an enhanced surrogate model with the following features:

- low-dimensional reparametrization of the vehicle geometry;
- incorporation of physical fields to enrich the data and raise the information content without additional CFD computations. This represents the biggest difference from traditional methods;
- reliance on physical formulas for calculating drag forces;
- ability to compute sensitivities, i.e. accuracy for small geometry variations.

Let us also emphasize that the "limited evaluations" constraint sets the work apart from methods relying on extensive datasets (like deep neural networks and related data-intensive ML approaches). In addition, the use of artificial data generative AI systems cannot meet the requirement of accuracy for slight geometric changes (final fine styling phase).

The paper is organized as follows. Section 2 is dedicated to the methodology of the method. It includes integral formulas of drag force (section 2.1), the shape encoding and reparametrization (section 2.2), the low-order representation of the momentum formula in the wake cutting plane (section 2.3), the definition of the drag surrogate model (section 2.4) and its use for a new 'query' shape (section 2.6). Section 3 gathers all the numerical experiments and validations being done. The paper will end up with concluding remarks and perspectives in Section 4.

1.6. Related works

This section is intended to give an overview of the recent literature and related works in the context of surrogate modeling and machine learning methods for automotive drag force prediction. In aerodynamic studies, the performance of a design is often evaluated based on the drag coefficient. Some models only predict this scalar coefficient, while others predict the entire flow field, from which the drag is computed. Both approaches require extensive input data to achieve high prediction accuracy. Two main categories of models emerge in the literature: those used in early-phase project development with a broad range of shape variations, and those designed for more specific geometries variations. The ShapeNet dataset ¹ is commonly employed for training models that need to handle diverse car shapes. For instance, [14], proposed a method using Gaussian Process with optimized weights to predict drag coefficient along with pressure and velocity field surrounding simplified 3D cars (without spoilers, tires, side mirrors) of this dataset. Car shapes are parameterized using a Polycube map, which allows for a fixed-length representation of all shapes. The positions of PolyCube mesh surface points are used as for the GP model. Trained with 889 shapes, the model the model's prediction accuracy improves as the training set size increases, with the error decreasing from 5.4% for 100 samples to 3.4% for 889 samples. Additionally, the model achieves an average Mean Squared Error of 0.48 on velocity and 8.1 on pressure when trained with the full dataset. A manifold based approach was presented in [19], where authors use a Locally Linear Embedding, to predict the drag of 3D car shapes. The method assumes that drag coefficient of a car can be obtained through linear combination of 5 neighboring cars in the manifold of car shapes. To build the manifold, the signed distance function of 60 car shapes from ShapeNet dataset was used. An overall average relative percent error of 11.5% is achieved with this approach. In [18] the authors applied a similar method to estimate the airflow around the vehicle using with a reduced input dataset. The model was constructed using the signed distance

¹ <https://shapenet.org/>

function 70 car shapes, including various types like sedans and sports cars. The model achieved an L^2 error on the velocity field ranging from 0.56% to 1%, depending on the car type. Neural Networks (NN), especially Convolutional Neural Networks (CNN) have been widely adopted due to their efficacy in extracting spatial features. For instance, [31] proposed an improved *ResNeXt* model [34] that uses 2D depth and normal renderings to represent 3D car geometries. By training on 9,070 shapes from ShapeNet, their model achieved an overall average drag prediction error of 2.4%. The Shapenet dataset is also used by [37] to train a coordinate-based multilayer perceptron to predict flow field around a 2D car profile. Trained on 1812 profiles, derived from the central vertical cross-section of 3D shapes of the ShapeNet dataset, the model achieves an average MSE of 7.15% for drag prediction. Similarly, [20] uses Convolutional Neural Networks on arbitrary 2D primitives to predict airflow around 2D car silhouettes. This model returns a prediction error of 15.34%. In [23], a U-Net [36] architecture is employed to predict drag coefficients and flow field of car shapes from the geometry [35]. This model uses a signed distance function for 3D geometry representation and is trained on up to 10,080 samples, achieving a maximum absolute error of approximately 2.5% in drag prediction and 2.8% on the L^2 error of field prediction. Alternatively, a Geodesic Convolutional Neural Network [38] is used in [21] to predict pressure and velocity distribution on the surface of a 3D car. As in [14], the car geometries are remeshed using a Polycube map for consistent representation. The model trained on approximately 2,000 randomly generated car shapes, report an accuracy of 51% for new prediction. The accuracy of the model can reach up to 70% when the model is training dataset is augmented with 54 real car shapes. This Geodesic Convolutional Neural Network is used in [39] in for drag optimization. The GCNN-based approach outperforms traditional methods like Kriging in creating a surface response for optimal drag coefficient prediction. However no details on the error or accuracy of the surface response is provided by authors. A optimization study is also presented in [32], where the authors introduce MeshSDF, a model designed for the generation and optimization of 3D geometries. The model learns from point cloud representations of the 3D geometry and uses a CNN to predict the pressure field over the shape. Trained on 1,400 car shapes from the ShapeNet dataset, the model is employed to generate optimized shapes that minimize pressure drag from an initial configuration. However, the authors provide no details on the accuracy of their pressure field predictions or the resulting drag estimations.

Another category of models focuses on studying specific geometric variations for fine styling adjustments. These models aim to predict drag or flow fields when a defined car geometry undergoes specific modifications. This is the focus of our paper, as such models are important for optimizing aerodynamic performance. Despite their importance, there is limited research on applying these types of models within the automotive industry. In [16], a mathematical model is built to generate new silhouettes of a sedan parameterized with 21 design parameters, and then estimate their drag coefficient. The mathematical model is constructed using a two-step procedure to predict the drag coefficient from the coordinates of 52 control points. First, Principal Component Analysis (PCA) is applied to reduce the number of variables used for the geometry representation. These principal components are then used in a linear regression model to predict the drag coefficient. The performance of the regression model for drag prediction is evaluated for different training sample sizes, ranging from 100 to 1000. As the input sample size increases, the mean absolute percentage error decreases from 13.86% to 9.49%. For the largest sample size, 70% of the predictions have an error below 10%. However, the error can still reach up to 65% for certain silhouette types. Scaling up to 3D geometries, a Radial Basis Function is used in [13] to map design shape parameters to drag coefficient of a 3D car with 12 shape parameters. Initially, with 164 high-fidelity evaluations from Latin hypercube sampling, the model achieves a 3.7% prediction error for the optimal shape. By using an Adaptive Multi-Scale RSM methodology, which refines the response surface in an iterative manner, the error is reduced to less than 1% with increased sample sizes (280 to 480). This approach also makes the model less sensitive to the number and distribution of input samples compared to traditional RSM

methods methodologies. Closest to our work, [15] proposes a model that combines Proper Orthogonal Decomposition (POD) reduction with kriging interpolation to evaluate pressure and velocity fields on the surface of a car. In their approach, POD is applied to the snapshot matrix of results from a Detached Eddy Simulation (DES) for a car shape with three parameters. A kriging model is then used to interpolate the POD basis coefficients based on shape modifications. The model is trained using multiple datasets with varying distributions and input sample sizes ranging from 8 to 38. As the number of input samples increases, the relative error decreases from 2.3% to 1.1% for pressure prediction and from 2.9% to 1.7% for viscosity prediction. With a Latin hypercube dataset of 20 samples, the mean error in drag prediction using the estimated fields is 0.3%. Additionally, a variable fidelity approach is explored to enhance the high-fidelity DES snapshots with lower fidelity Reynolds-Averaged Navier-Stokes (RANS) snapshots. However, this approach did not yield better prediction results for the test case considered.

The reviewed approaches highlight the use of extensive input data for developing effective models in automotive drag force prediction. They also reveal a notable gap in research concerning fine-tuning models for specific geometric variations.

2. Methodology

2.1. Drag force evaluation methods

Aerodynamic simulations of automotive vehicles are conducted at very high Reynolds numbers, typically of the order 10^7 involving thin boundary layers and developed turbulence. To address these high Reynolds number flows, the solver relies on the Large Eddy Simulation (LES) method. The LES is a numerical modeling approach aimed at directly capturing the large unsteady turbulent structures. Furthermore, it is combined with a Smagorinsky subgrid scale (SGS) model as closure model. This combination enables the representation of turbulent features at a smaller scale than the primary grid resolution, thereby enhancing the precision of the models. These two concurrent approaches empower the solver to faithfully represent turbulent flows across various scales.

The evaluation of the drag force f_x can then be carried out through the following equations [25,26]:

$$\mathbf{f} = \iint_{S_v} ((p - p_0) \mathbf{n} - \boldsymbol{\tau} \cdot \mathbf{t}) d\sigma, \quad f_x = \mathbf{f} \cdot \hat{\mathbf{x}} \quad (1)$$

where p represents the static pressure, p_0 denotes the infinite unperturbed pressure, $\hat{\mathbf{x}}$ is the unit vector in the x direction, $\boldsymbol{\tau}$ represents the viscous stress tensor, \mathbf{n} stands for the unitary normal vector on the vehicle's surface S_v , and \mathbf{t} is the tangential unit vector. The viscous stress tensor is calculated as $\boldsymbol{\tau} = \nu (\nabla \mathbf{u} + \nabla \mathbf{u}^T)$, with ν representing the kinematic viscosity and $\nabla \mathbf{u}$ denoting the velocity gradient tensor.

Rather than determining aerodynamic forces through the integration of stresses on the vehicle surface, the drag force can also be evaluated on another 'virtual' surface into the fluid domain. Assuming a bounding domain which is far enough from the vehicle, the drag force can be estimated on a fixed rear cutting plane S_{cp} which is orthogonal to $\hat{\mathbf{x}}$, located in the wake of the vehicle, according to the so-called far field formula, also known as Onorato formula [28]:

$$f_x = \iint_{S_{cp}} (p_0 - p + \rho [u_0 u_x - (u_x)^2]) dS. \quad (2)$$

Here, ρ represents the fluid density, u_0 is the free stream velocity, and u_x is the velocity component in the direction of interest x . This will allow us to get snapshot fields that are defined on the grid S_{cp} that will do not depend on the vehicle shapes. The surface drag coefficient is computed from the aerodynamic force f_x with the formula [26]: $C_d A = f_x / (\frac{1}{2} \rho_0 u_0^2)$.

Because of the use of a LES turbulence model (see [27]), the Navier-Stokes solver returns unsteady solutions. However, when evaluating vehicle performance, metrics such as drag coefficient $C_d A$ are usually searched as constant values. The LES-based Navier-

Stokes solutions are expected to reach a stationary flow after a transient phase, in the sense of a stationary stochastic process (stationary flow being averaged over a suitable time window). Thus, the drag coefficient $C_d A$ is typically computed from time-averaged quantities to return an expectation value. When computing a mean far field drag, a careful attention must be paid to evaluate mean values of nonlinear terms like $(u_x)^2$.

The shape-parametrized reduced-order surrogate model proposed in this paper is composed of three stages: i) shape encoding; ii) extraction of proper orthogonal modes on the cutting plane; iii) surrogate model of force density and drag force. In what follows, we give details of the three stages.

2.2. Shape encoding

In the context of shape design optimization in Automotive Engineering, the assessment of performance often requires the evaluation of numerous geometrical variations of the vehicle through a numerical design of experiment for example. To generate deformed geometries from a reference one, a direct morphing approach can be used. Local regions of reference geometry are transformed by defining a control area, where specific displacement are imposed, a free area, free to move; and a fixed area, which remain stationary during the morphing process. Assume that the shape of the vehicle depends on p CAD parameters $(\mu_k)_{k=1,\dots,p}$ that are gathered into a vector $\boldsymbol{\mu}$. The domain of admissible shape parameters $\boldsymbol{\mu}$ is denoted by $\mathcal{M} \subset \mathbb{R}^p$. In the sequel, the car shape for parameter $\boldsymbol{\mu}$ will be denoted by $\underline{S}(\cdot, \boldsymbol{\mu})$. It will be assumed that each shape $\underline{S}(\cdot, \boldsymbol{\mu})$ is a slight variation of a reference shape denoted by $\underline{S}^{ref}(\cdot) = \underline{S}(\cdot, \boldsymbol{\mu}^{ref})$.

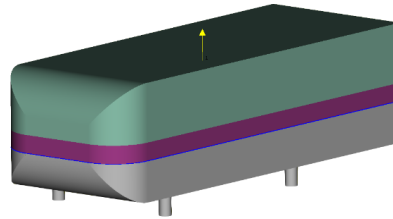


Figure 1. Morphing regions on the well-known Ahmed body with one CAD parameter acting on the global height. Control areas are represented in green, free area in purple and the boundary areas in blue.

To capture as much information as possible from the entire 3D shape and be able to discriminate different admissible shapes, it is important to have a more informative and discriminating descriptor, able to characterize the principal components of geometrical variation. For efficiency purpose, such a shape encoding has also to be set up in a rather low-dimensional space.

The procedure used in the present work is explained in the sequel. As it is assumed that the morphed geometry $\underline{S}(\cdot, \boldsymbol{\mu})$ is a slight variation from the reference surface $\underline{S}^{ref}(\cdot)$. It can be characterized by a shape displacement vector field $\underline{\delta}(\cdot, \boldsymbol{\mu}) \in [L^2(\underline{S}^{ref})]^3$, evaluated on the reference shape:

$$\underline{S}(\cdot, \boldsymbol{\mu}) = \underline{S}^{ref}(\cdot) + \underline{\delta}(\cdot, \boldsymbol{\mu}). \quad (3)$$

The idea is to approximate all the admissible displacements in a rather low-dimensional vector space defined by a suitable orthogonal reduced basis. This can be done for example from a method of snapshots of shapes.

Let $\{\underline{S}(\cdot, \boldsymbol{\mu}^k)\}_{k=1,\dots,N_s}$ be a set of N_s precomputed deformed shapes from a suitable CAD parameter sampling $\{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^{N_s}\}$. From these shape snapshots, it is easy to find an orthogonal basis $\underline{q}_\ell(\cdot) \in [L^2(\underline{S}^{ref})]^3$, $\ell = 1, \dots, K$, $K \leq N_s$, either by means of a principal component analysis (PCA) or more simply by means of a Gram-Schmidt (GS) orthogonalization procedure. If N_s is big enough, it is preferable to apply a PCA to find the K principal components, expecting that $K < N_s$. In the case of limited data with N_s small, one can

consider $K = N_s$ and simply use the GS algorithm (i.e. QR factorization in the discretized context). Thus, for a query parameter vector $\boldsymbol{\mu} \in \mathcal{M}$,

$$\underline{S}(\cdot, \boldsymbol{\mu}) \approx \underline{S}^{ref}(\cdot) + \sum_{\ell=1}^K v_\ell(\boldsymbol{\mu}) \underline{q}_\ell(\cdot). \quad (4)$$

The linear combination coefficients $(v_\ell(\boldsymbol{\mu}))_{\ell=1, \dots, K}$ provide a set of descriptors of the shape $\underline{S}(\cdot, \boldsymbol{\mu})$. They are computed from the orthogonal projection of $\underline{\delta}(\cdot, \boldsymbol{\mu})$ onto the vector space spanned by the basis functions $\underline{q}_\ell(\cdot)$:

$$v_\ell(\boldsymbol{\mu}) = (\underline{\delta}(\cdot, \boldsymbol{\mu}), \underline{q}_\ell)_{[L^2(S^{ref})]^3}. \quad (5)$$

The feature vector $\boldsymbol{v}(\boldsymbol{\mu})$ then serves as the new global shape descriptor to encode the geometry $\underline{S}(\cdot, \boldsymbol{\mu})$. This approach enables the creation of new shape descriptors that offer a more descriptive information than the CAD parameters. By leveraging the detailed information captured at the local level from local features, this method provides a comprehensive representation of the overall shape, resulting in a more informative and nuanced global shape descriptor set.

Discretized formalism

For computational purposes, any car shape is discretized thanks to a Finite Element (FE) triangular mesh as triangular elements are better suited for meshing complex geometries, such as detailed 3D car models. Let us assume a shape triangulation made of N_v vertices, $N_v \gg 1$. The FE displacement fields $\underline{\delta}(\cdot, \boldsymbol{\mu})$ as well as the orthogonal fields $\underline{q}_\ell(\cdot)$ are stored as large vectors $\boldsymbol{\delta}_\mu$ and \boldsymbol{q}_ℓ respectively. The shape encoding algorithm is the following:

1. Offline stage: assume that a snapshot database of shape displacements

$$D = [\boldsymbol{\delta}_{\mu^1}, \boldsymbol{\delta}_{\mu^2}, \dots, \boldsymbol{\delta}_{\mu^{N_s}}] \in \mathbb{R}^{3N_v \times N_s}$$

is available. From an SVD analysis or a QR factorization of D , compute an orthogonal reduced basis $\boldsymbol{q}_1, \dots, \boldsymbol{q}_K \in \mathbb{R}^{3N_v}$. Define the matrix

$$Q = [\boldsymbol{q}_1, \dots, \boldsymbol{q}_K] \in \mathbb{R}^{3N_v \times K};$$

2. Online stage: for a query CAD parameter $\boldsymbol{\mu}$, compute a mesh of the shape $\underline{S}(\cdot, \boldsymbol{\mu})$. Then compute the discrete displacement field $\boldsymbol{\delta}_\mu$ and the POD coefficients vector

$$\boldsymbol{v}(\boldsymbol{\mu}) = Q^T \boldsymbol{\delta}_\mu \in \mathbb{R}^K. \quad (6)$$

Remark that if N_s is small and the QR factorization $D = QR$ is used (in this case $K = N_s$), then

$$R = Q^T D \in \mathbb{R}^{K \times K}$$

is the matrix of the POD coefficients vector of each displacement vector $\boldsymbol{\delta}_{\mu^1}, \boldsymbol{\delta}_{\mu^2}, \dots, \boldsymbol{\delta}_{\mu^{N_s}}$ of D .

2.3. Knowledge extraction and reduced-order representation in the cutting plane

As already seen in the previous section, for a deformed shape $\underline{S}(\cdot, \boldsymbol{\mu})$, the aerodynamic drag can be estimated by integrating the physical scalar field

$$\kappa(\cdot, \boldsymbol{\mu}) \stackrel{\text{def}}{=} p_0 - p(\cdot, \boldsymbol{\mu}) + \rho [u_0 u_x(\cdot, \boldsymbol{\mu}) - u_x^2(\cdot, \boldsymbol{\mu})]$$

over a cutting plane S_{cp} using the far field formula: 359

$$C_d A(\boldsymbol{\mu}) = \frac{1}{\frac{1}{2}\rho_0 u_0^2} \iint_{S_{cp}} \kappa(\cdot, \boldsymbol{\mu}) dS_y. \quad (7)$$

The quantity $\kappa(\cdot, \boldsymbol{\mu}) \in L^2(S_{cp})$ can be seen as a drag force density per unit surface over the cutting plane S_{cp} . As each new shape is a slight deformation of the reference shape, the resulting flow field is also a variation of that of the reference configuration. Then the field force density $\kappa(\cdot, \boldsymbol{\mu})$ can be computed as the reference one plus a deviation $\Delta\kappa(\cdot, \boldsymbol{\mu})$: 360
361
362
363

$$\kappa(\cdot, \boldsymbol{\mu}) = \kappa^{ref}(\cdot) + \Delta\kappa(\cdot, \boldsymbol{\mu}). \quad (8)$$

Once again, from the different snapshot solutions $\{\kappa(\cdot, \boldsymbol{\mu}^k)\}_{k=1, \dots, N_s}$ previously computed with a high-fidelity CFD solver, one can extract an orthogonal reduced basis $\psi_\ell(\cdot) \in L^2(S_{cp})$, $\ell = 1, \dots, K$. That allows us to approximate the variation $\Delta\kappa(\cdot, \boldsymbol{\mu})$ in a low-dimensional vector space spanned by the orthogonal modes ψ_ℓ , i.e. 364
365
366
367

$$\Delta\kappa(\cdot, \boldsymbol{\mu}) \approx \sum_{\ell=1}^K b_\ell(\boldsymbol{\mu}) \psi_\ell(\cdot). \quad (9)$$

For the numerical implementation, the cutting plane is discretized using a Cartesian mesh or a Finite Element mesh composed of N_{cp} nodes. The FE discrete fields $\kappa(\cdot, \boldsymbol{\mu})$ as well as the orthogonal basis $(\psi_\ell(\cdot))_\ell$ are stored as large vectors $\boldsymbol{\kappa}_\mu$ and $\boldsymbol{\psi}_\ell$ respectively. The variation fields $\Delta\kappa(\cdot, \boldsymbol{\mu})$ are also stored as vectors $\Delta\boldsymbol{\kappa}_\mu = \boldsymbol{\kappa}_\mu - \boldsymbol{\kappa}^{ref}$. The algorithm is then the following: 368
369
370
371
372

1. Compute the snapshot matrix $U \in \mathbb{R}^{N_{cp} \times N_s}$ of field forces by collecting results of the high-fidelity CFD solver for the N_s training shapes: 373
374

$$U = [\Delta\boldsymbol{\kappa}_{\mu^1}, \dots, \Delta\boldsymbol{\kappa}_{\mu^{N_s}}]. \quad (10)$$

2. Extract the modes $\boldsymbol{\psi}_\ell \in \mathbb{R}^{N_{cp}}$, $\ell = 1, \dots, K$ by performing either a principal component analysis or a QR factorization of matrix U , depending of the number of snapshots. Then define the matrix 375
376
377

$$P = [\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_K] \in \mathbb{R}^{N_{cp} \times K}.$$

For very limited snapshot data, it is preferable to use a QR factorization. In this case we have $K = N_s$ and $U = PT$ with $T \in \mathbb{R}^{K \times K}$ an upper triangular matrix. Since the matrix P is semi-orthogonal, we have $T = P^T U$. 378
379
380

2.4. Parametric surrogate model 381

Equations (8),(9) invite us to search $\kappa(\cdot, \boldsymbol{\mu})$ in the form 382

$$\kappa(\cdot, \boldsymbol{\mu}) = \kappa^{ref}(\cdot) + \sum_{\ell=1}^K b_\ell(\boldsymbol{\mu}) \psi_\ell(\cdot)$$

for a new query vector $\boldsymbol{\mu}$, with corresponding shape $S(\cdot, \boldsymbol{\mu})$. One can then use the formula (7) to compute $C_d A(\boldsymbol{\mu})$. But of course the coefficients $b_\ell(\boldsymbol{\mu})$ are not known and a 'closure' is needed. 383
384
385

In the previous Section 2.2, we have considered a shape encoding defined by the coefficient vector $\boldsymbol{v} = \boldsymbol{v}(\boldsymbol{\mu})$. The partial information we have on the shape $S(\cdot, \boldsymbol{\mu})$ is described by $\boldsymbol{v}(\boldsymbol{\mu})$. So it is better to consider the combination coefficients b_ℓ as functions of $\boldsymbol{v}(\boldsymbol{\mu})$: 386
387
388
389

$$\kappa(\cdot, \boldsymbol{\mu}) = \kappa^{ref}(\cdot) + \sum_{\ell=1}^K b_\ell(\boldsymbol{v}(\boldsymbol{\mu})) \psi_\ell(\cdot). \quad (11)$$

The coefficients $b_\ell(\mathbf{v}(\boldsymbol{\mu}))$ can be seen as nonlinear features of $\mathbf{v}(\boldsymbol{\mu})$. But actually these functions are unknown. In the context of limited data, a large model with multiple degrees of freedom such as a DNN is irrelevant because of the lack of data. The simplest reasonable 'closure' one can consider in this context is a linear dependency of each $b_\ell(\mathbf{v}(\boldsymbol{\mu}))$ with $\mathbf{v}(\boldsymbol{\mu})$, meaning $b_\ell(\mathbf{v}(\boldsymbol{\mu}))$ chosen in the form

$$b_\ell(\mathbf{v}(\boldsymbol{\mu})) = \sum_{m=1}^K a_{\ell m} v_m(\boldsymbol{\mu}). \quad (12)$$

The $\Delta\kappa$ model proposed in this paper is

$$\Delta\kappa(\cdot, \boldsymbol{\mu}) \approx \sum_{\ell=1}^K \sum_{m=1}^K a_{\ell m} v_m(\boldsymbol{\mu}) \psi_\ell(\cdot) \quad (13)$$

with a constant coefficient matrix $A = (a_{\ell m})_{\ell, m} \in \mathbb{R}^{K \times K}$. The matrix has to be identified to fit the available data. This can be done for example from the available sampled data $\{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^{N_s}\}$ by solving the least square problem

$$\min_{A \in \mathbb{R}^{K \times K}} \frac{1}{2} \sum_{k=1}^{N_s} \left\| \Delta\kappa(\cdot, \boldsymbol{\mu}^k) - \sum_{\ell=1}^K \sum_{m=1}^K a_{\ell m} v_m(\boldsymbol{\mu}^k) \psi_\ell(\cdot) \right\|_{L^2(S_{cp})}^2. \quad (14)$$

From the FE discretized point of view, the minimization problem (14) reads

$$\min_{A \in \mathbb{R}^{K \times K}} \frac{1}{2} \|U - PAR\|_F^2$$

with $R = [\mathbf{v}(\boldsymbol{\mu}^1), \dots, \mathbf{v}(\boldsymbol{\mu}^{N_s})] \in \mathbb{R}^{K \times N_s}$. The first order optimality conditions give the solution

$$A = \underbrace{P^T}_{K \times N_{cp}} \underbrace{U}_{N_{cp} \times N_s} \underbrace{R^T}_{N_s \times K} \underbrace{(RR^T)^{-1}}_{K \times K}$$

assuming that R has a maximal rank so that RR^T is invertible. In the case $K = N_s$ and under the maximal rank assumption, using the QR factorization $U = PT$, the solution simply writes

$$A = P^T U R^{-1} = T R^{-1}.$$

The discretized version of (11),(12) is

$$\boldsymbol{\kappa}(\boldsymbol{\mu}) = \boldsymbol{\kappa}^{ref} + PA\mathbf{v}(\boldsymbol{\mu}), \quad (15)$$

which is equivalent to

$$\boldsymbol{\kappa}(\boldsymbol{\mu}) = \boldsymbol{\kappa}^{ref} + \sum_{\ell=1}^K \sum_{m=1}^K a_{\ell m} v_m(\boldsymbol{\mu}) \boldsymbol{\psi}_\ell. \quad (16)$$

Finally, the surface drag is computed from the formula (7):

$$C_d A(\boldsymbol{\mu}) = \frac{1}{\frac{1}{2} \rho_0 u_0^2} \iint_{S_{cp}} \boldsymbol{\kappa}(\cdot, \boldsymbol{\mu}) dS_y. \quad (17)$$

In practice, the integral is approximated using a FE quadrature on each each FE element over the cutting plane. We have $C_d A(\boldsymbol{\mu}) = \langle \mathbf{w}, \boldsymbol{\kappa}(\boldsymbol{\mu}) \rangle$ for some vector \mathbf{w} .

2.5. Online stage: drag force evaluation

For a new parameter vector $\boldsymbol{\mu}$, the deformed surface is created and is discretized by a triangular mesh. If the mesh does not share the same connectivity as the reference shape mesh, a mesh matching method such as transport optimal (Appendix A) has to be used.

The set of shape feature $\nu(\mu)$ is computed by formula (6). Then the force density $\kappa(\mu)$ is computed from Equation (16). Finally, the drag $C_d A(\mu)$ is assessed by means of the formula (17).

2.6. Summary

The offline and online stages of the proposed surrogate model are summarized below in Algorithms 1 and 2 below.

Algorithm 1 ROM far field Offline Phase - Learning phase for a set of N_s training shapes $\{S(\cdot, \mu^k)\}_{k=1, \dots, N_s}$ with FOM results

Require: Reference geometry \underline{S}^{ref} and deformed shapes $\{S(\cdot, \mu^k)\}_{k=1, \dots, N_s}$

Ensure: Q, ψ, A

- 1: Discretize shapes by a mesh of N_v vertices
 - 2: Compute snapshot matrix of displacements $D = [\delta_{\mu^1}, \delta_{\mu^2}, \dots, \delta_{\mu^{N_s}}]$
 - 3: Compute orthogonal reduced basis $Q = [q_1, \dots, q_K]$ by SVD analysis or QR factorization of D
 - 4: Compute new global shape descriptors $\nu(\mu^k) = Q^T \delta_{\mu^k}$
 - 5: Compute snapshot matrix of field variations $U = [\Delta \kappa_{\mu^1}, \Delta \kappa_{\mu^2}, \dots, \Delta \kappa_{\mu^{N_s}}]$
 - 6: Extract orthogonal modes ψ by SVD analysis or QR factorization of U
 - 7: Compute coefficient matrix A
-

Algorithm 2 ROM Farfield Online Phase - Prediction of a new 'query' shape $\underline{S}(\cdot, \mu), \mu \in \mathcal{M}$

Require: Reference geometry \underline{S}^{ref} , deformed geometry $\underline{S}(\cdot, \mu)$, farfield modes ψ , coefficients matrix A , shape basis matrix Q , reference force density κ^{ref}

Ensure: Aerodynamic drag coefficient $C_d A(\mu)$

- 1: Discretize deformed shape by a mesh of N_v vertices
 - 2: Compute discrete displacement field δ_{μ}
 - 3: Compute shape descriptor $\nu(\mu) = Q^T \delta_{\mu}$
 - 4: Compute variation field $\kappa(\mu) = \kappa^{ref} + \sum_{\ell=1}^K \sum_{m=1}^K a_{\ell m} \nu_m(\mu) \psi_{\ell}$
 - 5: Compute drag coefficient $C_d A$
-

3. Numerical experiments, results and discussion

3.1. High Fidelity simulation

A CFD numerical simulation is carried out using the commercial solver *ProLB*, based on the Lattice Boltzmann Method (LBM). The LBM is a numerical approach to model macroscopic fluid dynamics by simulating the microscopic particle dynamics and their collisions on a discretized lattice grid. This method well handles complex geometries. To deal with turbulent flows near the vehicle, a LES model along with a LBM-based SGS (subgrid Smagorinski) model is used. The computational domain and boundary conditions are set to accurately represent a realistic wind tunnel setup. The domain is set 48.96 m long, 29.4 m high and 46.4 m wide. At the inlet of the flow domain, a realistic upstream velocity $u_0 = 45.88 \text{ m}\cdot\text{s}^{-1}$ is used, corresponding to a Reynolds number of order 10^7 . The atmospheric pressure $p_0 = 101,325 \text{ Pa}$ is imposed at the outlet. On all other bounding surfaces of the domain, a friction-less boundary condition is employed. Finally, on vehicle surfaces, a no-slip boundary condition is imposed. The computational domain is composed of eight successive Adaptive Mesh Refinement (AMR) resolution domains, corresponding to refinements of the LBM mesh. The finest resolution is of the order of

2 mm, near the vehicle to ensure accurate flow results. The Boltzmann equation is solved over the LBM mesh for a total of 520,000 iterations, which corresponds to 2.5 seconds of air flow simulation. This high number of iterations is set to ensure that the flow reaches a steady state. Pressure and velocity fields are captured on the lattice grid for the steady state iterations. The simulation takes approximately 7 hours on a cluster of 900 CPUs to complete.

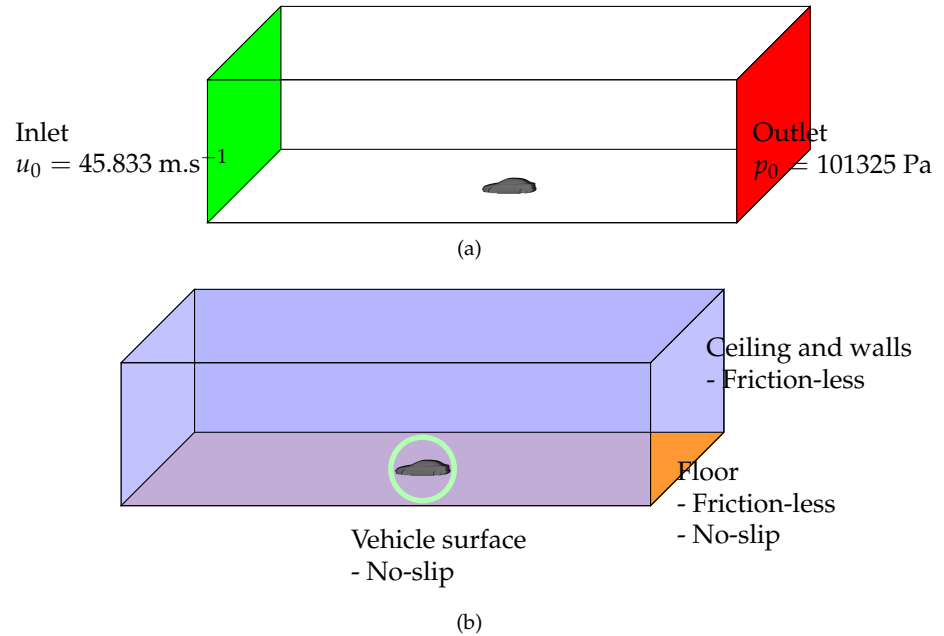


Figure 2. Boundary conditions CFD simulation 3D vehicle. **(a)** Boundary Conditions on inlet and outlet surfaces of wind tunnel. **(b)** Boundary conditions on wind tunnel ceiling, floor, lateral walls and vehicle

3.2. Simplified geometry "S2A"

The surrogate model has been tested on a simplified industrial car design, which is used in the Renault wind tunnel facility for aerodynamic calibration. This design has the essential structures of a real vehicle while having enough simplicity to reduce computational complexity. It is made of multiple detailed structures: a superstructure, an underbody, front and rear axles, tires, and rims. The vehicle geometry is simplified with no external appendices (such as external mirrors) or engine compartments. Parametrization of the car shape involves three independent parameters, i.e. $\mu = (\mu_1, \mu_2, \mu_3)$, affecting only the vehicle body superstructure. The first parameter, μ_1 , corresponds to a translation along the \hat{y} axis, simulating a narrowing of the rear bumper and constrained within the range of $[-20 \text{ mm}, 20 \text{ mm}]$. The second parameter, μ_2 , represents a translation along the \hat{x} axis to model an extension of the vehicle's roof, with a range of $[-50 \text{ mm}, 50 \text{ mm}]$. Lastly, the third parameter, μ_3 , denotes a rotation around the \hat{y} axis of the rear underbody, constrained within $[-5^\circ, 5^\circ]$. The specific morphing area impacted by these parameters is illustrated in Figure 3. The initial geometry is represented by a FE mesh composed of $N_v = 556\,354$ nodes.

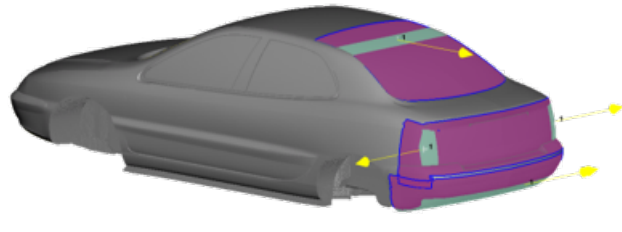


Figure 3. Presentation of the S2A geometry involving three independent parameters. The variable surface part of the geometry is drawn in magenta.

3.3. Data generation and preprocessing

A total of 30 deformed shapes are generated from the initial geometry, employing a maxi-min space-filling strategy on the CAD parameter μ (Figure 8). CFD simulations are performed for each shapes, and the high-fidelity results provide detailed information on the flow field around the vehicles and the corresponding drag coefficients (Table 1). The mean dimensionless surface drag of this dataset is 0.9952 with a standard deviation of 0.0127. The low standard deviation, compared to the mean, suggests that the drag coefficients are relatively consistent across the dataset since shapes variations are within a maximum range of 50 mm. However, in a design optimization study, even a small reduction in drag of 0.02 m^2 , is important as it could lead to a reduction of 1 gram of CO_2 consumption when proceeding to the Worldwide harmonized Light vehicles Test Procedure (WLTP) cycle.

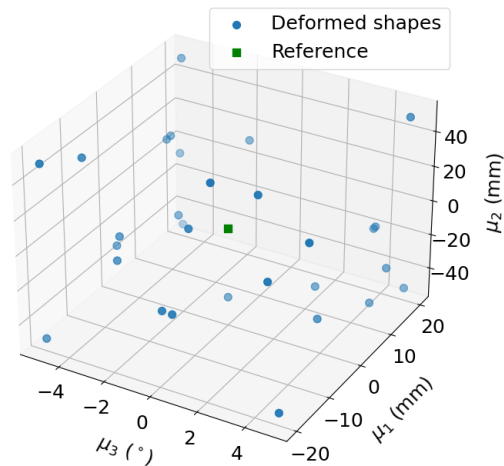


Figure 4. Maximin-type space-filling on CAD parameter vector $\mu \in \mathbb{R}^3$.

Table 1. Table of dimensionless drag results (the surface drag of the reference shape is 1). The range is [0.9632, 1.0168].

Shape ID	$\frac{C_d A}{C_d A^{ref}}$	Set	Shape ID	$\frac{C_d A}{C_d A^{ref}}$	Set
1	1.0000	Reference	17	1.0075	Training
2	0.9820	Validation	18	0.9974	Validation
3	0.9939	Validation	19	1.0168	Training
4	0.9836	Validation	20	0.9632	Training
5	1.0011	Validation	21	0.9616	Training
6	1.0046	Validation	22	0.9987	Validation
7	0.9994	Validation	23	0.9761	Validation
8	0.9831	Validation	24	0.9790	Validation
9	1.0088	Validation	25	0.9952	Validation
10	0.9921	Validation	26	0.9858	Validation
11	1.0054	Validation	27	0.9712	Validation
12	0.9844	Validation	28	0.9943	Validation
13	0.9785	Validation	29	1.0101	Validation
14	0.9955	Validation	30	1.0043	Validation
15	0.9969	Validation	31	0.9775	Validation
16	1.0100	Validation			

In this test case, each deformed geometry is independently meshed, therefore the meshes do not share the same mesh topology as the reference shape. As explained in Appendix A, an optimal transport strategy is used to make the mapping and mesh correspondence. The optimal transport problem was solved by using the reference open-source Python library `Geomloss` [29].

3.4. Model performance

The performance of the model with respect to the number of input data points and their distribution is evaluated. To achieve this, cross-validation is performed using different numbers of folds. The number of folds, denoted by N_f , determines how the data is split for training and validation. The dataset is divided into N_f subsets of approximately equal size. Usually in cross-validation, for each round, one subset is used for validation while the remaining $N_f - 1$ subsets are used for training. However in this study, as we aim at keeping a low number of training data, the typical approach is reversed : one fold is used for training and the remaining folds are used for validation. This allow to asses how well the model performs when it is trained on a small dataset. To ensure robustness of cross-validation results, multiple cross-validation runs are performed for each choice of fold number N_f . In each run, a different random seed is used to shuffle the data, resulting in different splits. This is done to minimize the impact of any particular split that may be suboptimal for the model. After performing these cross-validation runs, the results are averaged to give an estimate of the model's performance. This approach is especially important when the number of folds is small, as the randomness in splitting the data can sometimes lead to unfavorable training datasets, potentially leading to misleading performance evaluations. By repeating the process with different splits, the model's evaluation becomes more accurate and less dependent on any single data split.

Cross-validation is performed with the number of folds, N_f , varying between 2 and 8. This results in training subsets containing between 15 samples (for $N_f = 15$) and 3 or 4 samples (for $N_f = 8$). For each number of folds, a total of 10 cross-validation runs are executed.

First, the performance of the low-dimensional reparametrization of the vehicle geometry is evaluated for each cross-validation fold. To do so, the Mean Squared Error (MSE), interpreted here as the L^2 projection error, is computed for each shape in the validation datasets. This error measures how well the geometry of the vehicle is reconstructed from the reduced parameter space. The MSE is then averaged across all validation shapes for each fold. As shown in Figure 5 the average MSE tends to increase as the number of folds

increases as there is less data for training. However, the MSE values remain relatively low (less than 0.5 mm), even with a high number of folds. This demonstrates the effectiveness of the reparametrization in capturing the geometric variations of the vehicle shapes, even with limited training data.

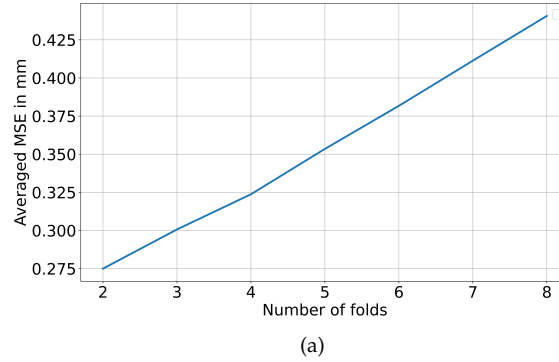


Figure 5. Averaged Mean Squared Error of shape reconstruction depending on the number of folds in cross-validation.

Then, to evaluate the performance of the model in predicting the drag coefficient for new shapes, several metrics are computed during cross-validation. First, the maximal and mean absolute error on $C_d A$ prediction are assessed. Figure 6(a) show the results for the different folds. Although the error tends to increase slightly with the number of folds, the overall error remains well within an acceptable threshold of 3%, indicating robust performance of the model with limited input data. The Pearson coefficient is computed to quantify the linear correlation between the predicted and high fidelity drag values (Figure 6(b)). The model consistently achieved high Pearson correlation coefficients across all folds. Since the model is intended to serve as a decision support tool during design optimizing process, it is important that the ranking of shapes regarding the drag is preserved. The model's ability to maintain the correct ranking is then evaluated using the Kendall rank correlation coefficient [30]. The Kendall coefficient is defined by the following equation [33]:

$$\tau = \frac{n_c - n_d}{n}, \quad (18)$$

with n_c the number of concordant pairs, n_d the number of discordant pairs and n the total number of pairs. A concordant pair is pair of observation where the predicted and high fidelity values have the same relative ordering. So, if one shape has a higher predicted drag value than another, and the actual drag values follow the same order, the pair is considered concordant. On the other hand a discordant pair is when the relative ordering is not preserved by the prediction. However this definition of discordant pairs can be too strict for aerodynamic studies. In some cases, pairs are considered discordant when the predicted and actual rankings are inverted, even though the difference in drag values between the configurations is very small. In such cases, the small deltas may not significantly affect the overall ranking for design studies. Therefore it would be more appropriate to classify these pair as concordant. A relaxed methodology is introduced to take into account this threshold. It introduces a threshold to account for minor prediction errors, allowing pairs with small simulated delta to remain concordant if the predicted delta is similarly close. For larger difference in drag, a more relaxed threshold is used. In the following analysis two variants of the relaxed Kendall coefficient are used, to assess the model ranking performance with different tolerance level. The first tolerance level allows a higher margin of error in the drag prediction, representing the maximum value for the prediction to be considered acceptable. The second tolerance level is stricter, requiring a smaller margin of error to classify the prediction as accurate. These two variants will be referred to as Relaxed

Kendall Acceptable and Relaxed Kendall Accurate. The tolerance for the Relaxed Kendall Acceptable coefficient is defined as:

$$threshold_{acceptable} = \begin{cases} |\Delta C_d A| + \frac{|\Delta C_d A|}{5} + 0.009 & \text{if } |\Delta C_d A| > 0.005 \\ |\Delta C_d A| + 0.01 & \text{otherwise} \end{cases} \quad (19)$$

For the Relaxed Kendall Accurate coefficient it is defined as:

$$threshold_{accurate} = \begin{cases} |\Delta C_d A| + \frac{|\Delta C_d A|}{5} + 0.003 & \text{if } |\Delta C_d A| > 0.01 \\ |\Delta C_d A| + 0.005 & \text{otherwise} \end{cases} \quad (20)$$

Where $|\Delta C_d A|$ is the absolute drag difference between two configurations. A predicted drag difference $\Delta \tilde{C}_d A$ is classified as concordant if it has the same sign as $\Delta C_d A$ or if:

- $|\Delta \tilde{C}_d A| \leq threshold_{accurate}$ for the Relaxed Kendall Accurate coefficient, or
- $|\Delta \tilde{C}_d A| \leq threshold_{acceptable}$ for the Relaxed Kendall Acceptable coefficient.

The values of the Kendall coefficients depending on the number of folds are presented in Figure 6(b). The Kendall rank coefficient computed across all folds is in a range of 0.575 to 0.59. This low value is due to many pairs being classified as discordant with the strict initial definition of concordance and discordance. However, when the Relaxed Kendall is applied, using the modified definitions that account for minor prediction errors, a significant improvement in the score is observed. Indeed across all folds the Relaxed Kendall Acceptable and Relaxed Kendall Accurate are in a respective range of 0.93 to 0.94 and 0.86 to 0.89.

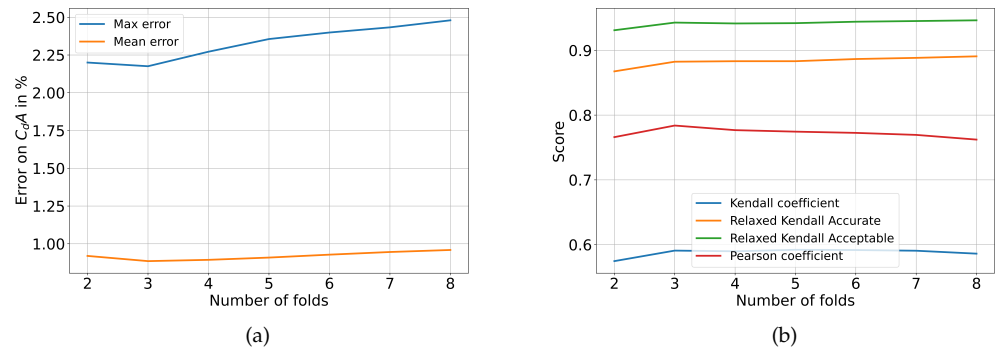


Figure 6. (a) Mean and max absolute relative error on $C_d A$ depending on the number of folds for cross-validation. (b) Correlation scores depending on the number of folds for cross-validation

The total time of the modeling process is measured, including the three main phases: data pre-processing, model training, and inference. The pre-processing of the data consists in mesh correspondence by using optimal transport. During this step, the computational time mainly depends on the number of triangular elements representing the shape. This step can be skipped if the mesh is morphed simultaneously with the geometry. When using the `Geomloss` library, optimal transport for a shape with $N_v = 556354$ elements takes approximately 15 minutes on a system with 40 CPUs. The training time depends on the number of input samples, which is linked to the number of folds in cross-validation. Figure 7 illustrates how the training time varies with different numbers of folds. As the number of folds increases, the training time decreases because the model becomes easier to solve with fewer data points in each training subset. The training of the model with 15 shapes (2 folds) takes approximately 10 seconds, while the training time drops to under 5 seconds when only 3-4 shapes (8 folds) are used in the training set. For the inference of a new shape, once the new geometry is generated using a morphing tool, no additional pre-processing via optimal transport is required if the mesh shares the same topology as the reference shape.

In this case, the inference time is extremely short, taking less than 1 second to obtain the results on the cutting plane and the drag force. 569
570

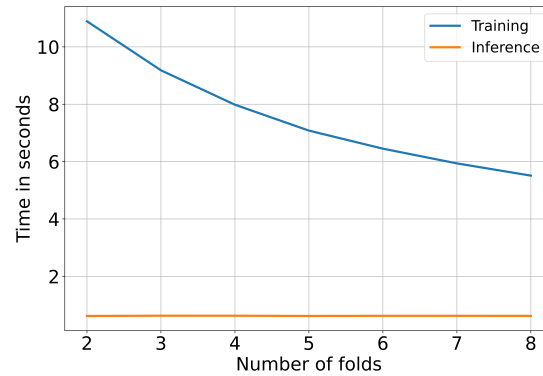


Figure 7. Training and inference time depending on the number of folds for cross-validation 571

Cross-validation experiment has demonstrated that, with a small number of folds, resulting in a large amount of input data, the performance of the model increases. However, even with limited training data, the performance of the model is relatively high. Therefore, it is advantageous to work with the smallest number of input data points, as this reduces the number of CFD simulations required. 572
573
574
575

3.5. Surrogate model construction 576

When working with a small input dataset, the choice of the training input data is very important to achieve a good accuracy. Therefore, in the following analysis four deformed shapes are chosen wisely (ID number 17, 19, 20, and 21 in Table 1) to train the model. The $N_s = 4$ input samples are selected to ensure a minimal comprehensive coverage of the exploration space of global parameters μ . Additionally the initial reference shape (shape ID 1) is included for computation of the shape variations. All the other shapes in the dataset serve as the validation set. The distribution of the selected shapes can be seen in Figure 8. 577
578
579
580
581
582
583

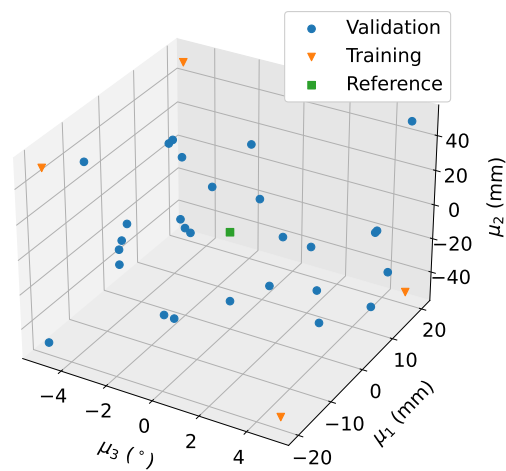
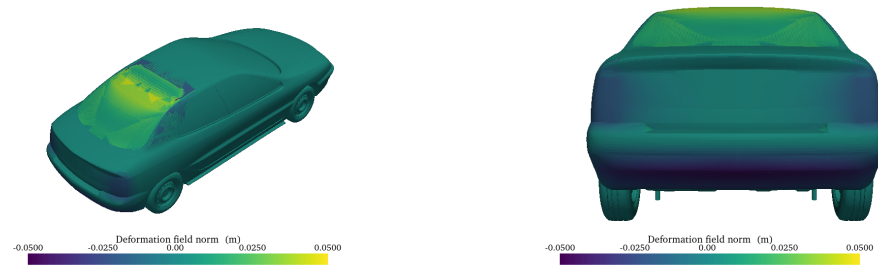


Figure 8. Maximin space-filling on CAD parameter $\mu \in \mathbb{R}^3$. Orange dots are the $N_s = 4$ training shapes, the green dot is the reference geometry $\mu = \mathbf{0}$ and the remaining blue dots are used for the validation set. 584

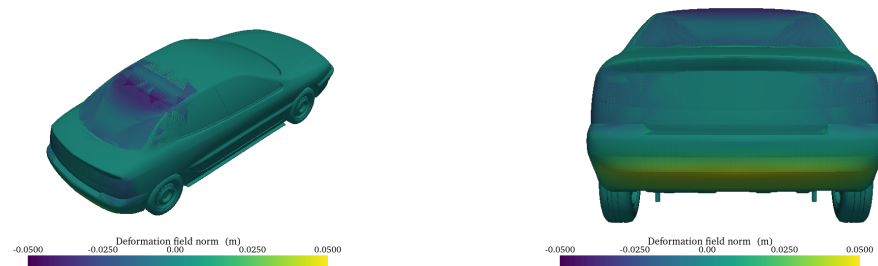
3.5.1. Shape encoding

Once the mesh mapping is done, shape variation fields $\delta_\mu \in \mathbb{R}^{3N_v}$ of the training geometries are used as shape variation basis. For information, the norm of the deformation field is shown in Figure 10 for two shapes of the training set. The training shapes cover both positive and negative deformations for parameters μ_1 , μ_2 and μ_3 , ensuring a faithful representation of the exploration space.



(a) Shape deformation field norm for training shape ID number 19 - Isometric view (b) Shape deformation field norm for training shape ID number 19 - YZ plane view

Figure 9. Shape deformation field norm for training shape ID number 19



(a) Shape deformation field norm for training shape ID number 21 - Isometric view (b) Shape deformation field norm for training shape ID number 21 - YZ plane view

Figure 10. Shape deformation field norm for training shape ID number 21

The field of δ_μ is used to compute new shape basis q_ℓ and resulting shape coefficients $v(\mu)$. The relative L^2 error projection on the given basis is plotted in Figure 11 for the validation shapes. The overall error is quite small, with a maximum error of 0.002 m for all shapes in the validation database.

This new basis enables good reconstruction of the shape, indicating that it effectively captures the essential features and variations present in the original data.

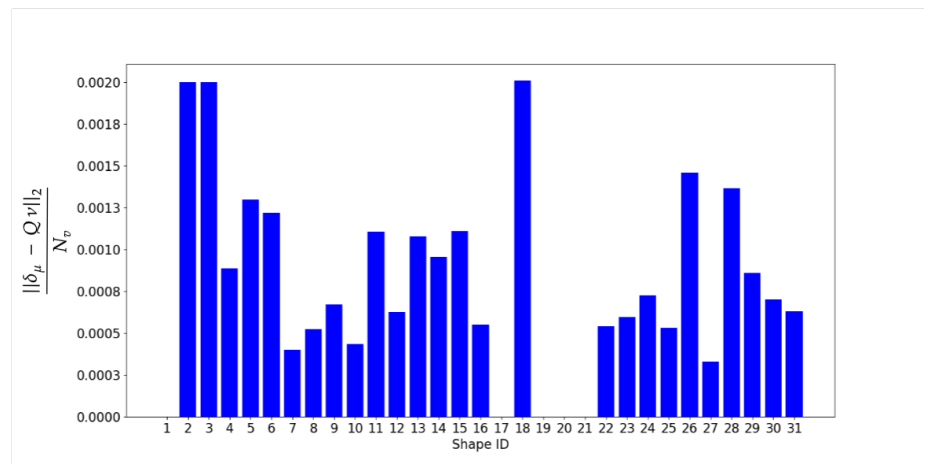
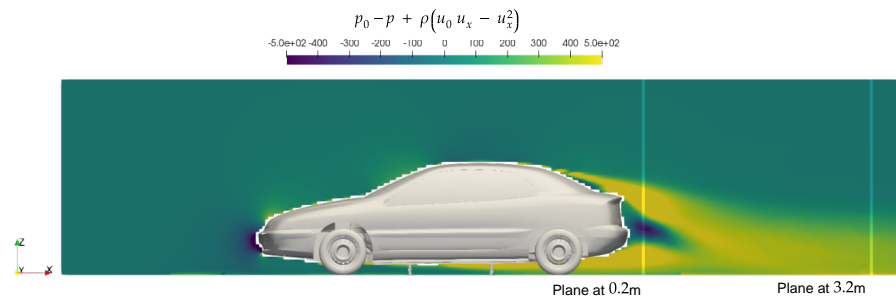


Figure 11. Relative L^2 error projection on the q_ℓ POD basis for each shape of the database.

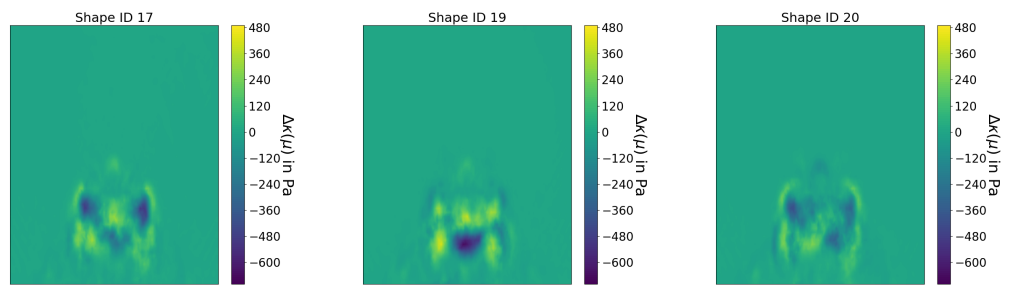
3.5.2. Computation of the flow field modes on a wake cutting plane

The snapshot matrix U of $\Delta\kappa$ fields is built from the FOM results. These results are obtained on a cutting plane located at a distance of 0.2 m behind the car (Figure 12(a)), discretized with $N_{cp} = 6930$ nodes. It is worth noting that the flow on a wake plane close to the vehicle is turbulent with large unsteady eddies and lacks the stabilization observed in planes situated farther away. Therefore, the influence of shape variations becomes more pronounced and discriminating on proximity planes, providing valuable insights of the aerodynamic behavior of the vehicle. As shown in Figure 12(b), 12(c) and 12(d) the flow field for the training shapes have different turbulent patterns. This difference will allow the model to learn a broader range of physical behavior, to enhance its ability of generalization for a better predictive performance.

596
597
598
599
600
601
602
603
604
605
606



(a) Cutting planes of study in the wake, at respective locations 0.2 m and 3.2 m from the vehicle.



(b) Delta fields $\Delta\kappa(\mu)$ for shape ID 17 on the wake plane located at 0.2 m from the vehicle. (c) Delta fields $\Delta\kappa(\mu)$ for shape ID 19 on the wake plane located at 0.2 m from the vehicle. (d) Delta fields $\Delta\kappa(\mu)$ for shape ID 20 on the wake plane located at 0.2 m from the vehicle.

Figure 12. Flow field on cutting plane located at 0.2 m from the vehicle

Far field modes, denoted as ψ_1 , ψ_2 , ψ_3 and ψ_4 , are computed and depicted in Figure ?? . Unlike Principal Component Analysis (PCA), the first mode resulting from the QR decomposition of U is not a linear combination of the columns of matrix U . This initial mode is the normalized first column of matrix U , as illustrated in Figure 13(a). Then each additional mode is orthogonal to the previous ones, so it captures distinct and independent flow field dynamics. With this approach, each mode provides new insights on the pattern of the flow behavior.

607
608
609
610
611
612
613

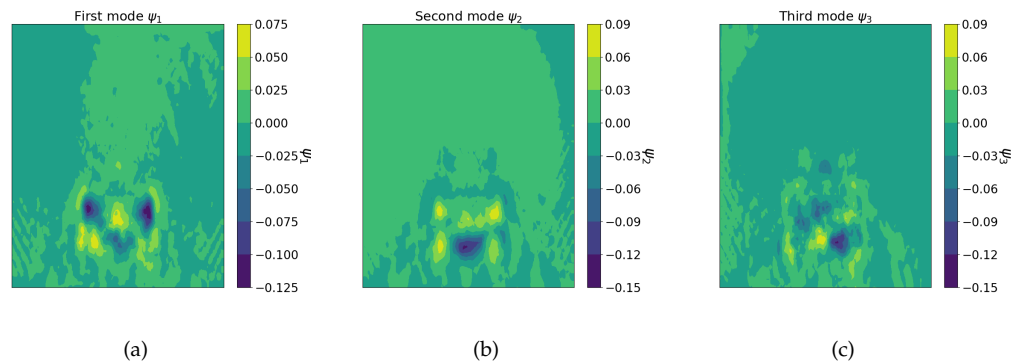


Figure 13. Modes on cutting plane S_{cp} . (a) Contour plot of far-field modes ψ_1 (b) Contour plot of far-field mode ψ_2 , (c) Contour plot of far-field mode ψ_3

3.6. Surrogate model evaluation

Once the reparametrization has been completed and the physical modes are computed, the constant coefficient matrix A is determined, which concludes the model training on the $N_s = 4$ shapes. The model is then applied to predict the density force $\kappa(\mu)$ on the cutting plane for the deformed shapes in the validation dataset. Figure 14 compares the flow field from the high-fidelity simulation with the predicted one from the surrogate model for one arbitrary shape from the validation set (Shape ID 7). The model successfully captures the overall flow patterns, confirming that the surrogate model accurately represents the fluid behavior. However, as shown in Figure 14(c) errors are observed in some turbulent regions, where the model struggles to capture finer details.

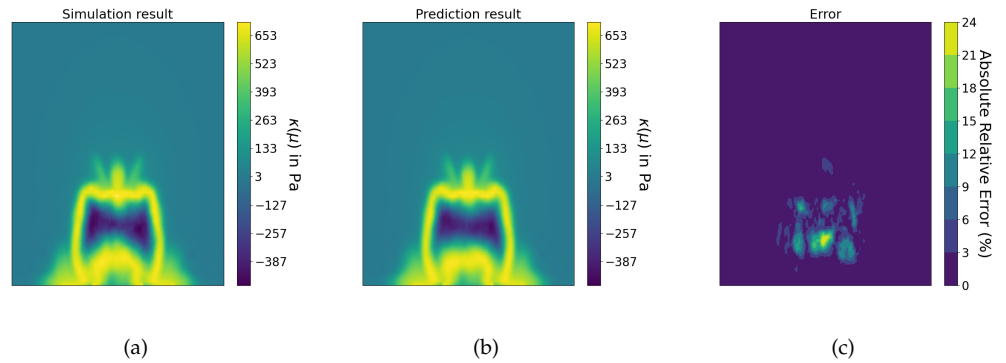


Figure 14. Prediction results of the flow field (a) Simulated flow field (b) Predicted flow field (c) Absolute relative error

The MSE of the flow field predictions is given in Figure 15 for the different shapes of the validation dataset. The MSE value is relatively low for all configurations with an average MSE of 0.38, indicating a strong overall predictive performance of the model.

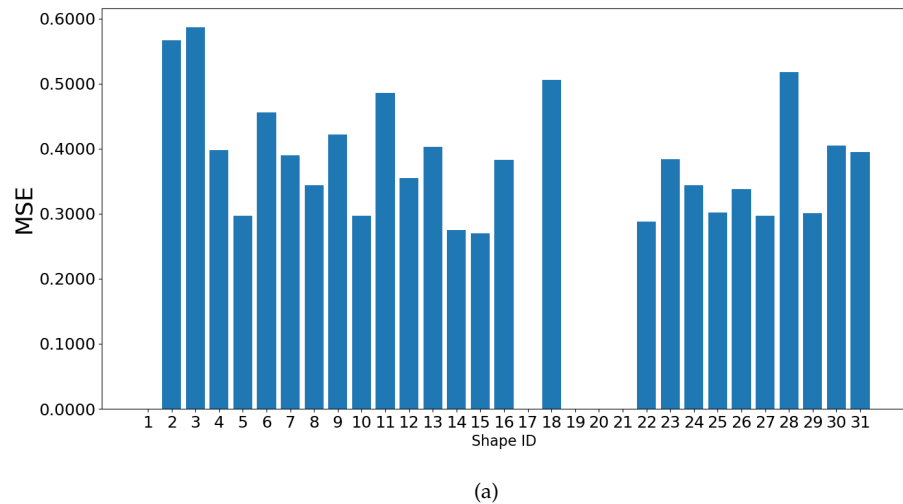


Figure 15. Mean Squared Error of the flow field predictions for various shapes in the validation dataset.

The predicted flow field density $\kappa(\mu)$ is integrated over the cutting plane by the mean of Equation 17 to obtain the drag coefficient of each deformed shape. The accuracy of the surrogate model is assessed by comparing the predicted drag coefficient against those obtained from FOM results. As shown by the correlation plot in Figure 16(a), there is a strong positive correlation between predicted and real drag coefficients, with a Pearson coefficient of 0.86. All predicted coefficients lie in the admissible threshold error range of

0-3 % (green dotted line), with a maximal error of 1.59 %. However, it's important to note that the order is not perfectly preserved for some geometries, indicating that the model captures the overall trend but may struggle with specific variations. To further examine these nuanced behaviors at particular points, the correlation plot of delta drag between each configuration is presented in Figure 16(b). This plot reveals a positive correlation between predicted and real delta. Most points are located within the threshold for acceptable (red dotted line) and accurate (green dotted line) prediction. More than 91% of delta between configurations are predicted accurately with only 2% of prediction outside the acceptable range. Moreover, the associated Kendall coefficients for this model are as follows : 0.67 for the standard Kendall coefficient, 0.95 for the Relaxed Kendall Accurate and 0.99 for the Relaxed Kendall Acceptable . Those different results demonstrate the model's excellent effectiveness in predicting drag coefficient variations.

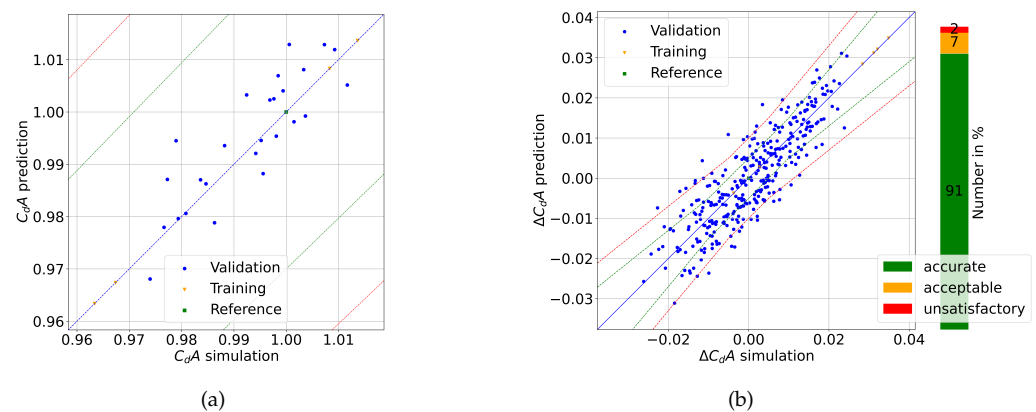


Figure 16. Prediction results (a) Correlation plot of $C_d A$. (b) Correlation plot of delta of $\Delta C_d A$ and its distribution.

4. Concluding remarks and perspectives

In this paper, we presented a novel ML-based surrogate model for predicting drag forces on shape-parametrized cars, addressing the constraints of limited high-fidelity CFD simulations. The model features low-dimensional reparametrization of vehicle geometry, integration of physical fields to enhance data information without extra CFD computations, reliance on physical drag formulas, and accuracy for small geometry variations.

The surrogate model showed robust predictive capabilities across various 3D deformed geometries, enabling rapid assessment of minor shape variations during optimization. The model achieves typical mean square errors of order 2 % for drag prediction with only few evaluations. While occasional challenges in preserving variation order were noted, the consistently low maximal error underscores its reliability for decision support and optimization studies.

Future work aims to enhance the model by incorporating more physical field modes and exploring different deformation fields for reparametrization. Additionally, refining the Design of Experiment selection could further optimize the methodology.

Author Contributions: Conceptualization, K.N.C., F.D.V. and Y.G.; methodology, K.N.C., F.D.V. and Y.G.; software, K.N.C. and Y.G.; validation, K.N.C. and Y.G.—original draft preparation, K.N.C.; writing—review and editing, K.N.C., F.D.V. and Y.G.; visualization, K.N.C.; supervision, F.D.V. and Y.G.; project administration, Y.G.; funding acquisition, Y.G. . All authors have read and agreed to the published version of the manuscript.

Funding: This material is based upon work supported by the french ANRT (Association Nationale de la Recherche et de la Technologie) with a CIFRE Fellowship (2020/1218) granted to Renault Group. 666
667

Data Availability Statement: Data is unavailable due to privacy restrictions. 668

Acknowledgments: The authors would like to thank Dr. Yves Tourbier for fruitful discussions toward this topic. 669
670

This work was performed using HPC resources from GENCI-IDRIS (Grant 2023-AD011012779R1) 671

Conflicts of Interest: The authors declare no conflicts of interest. 672

Abbreviations 673

The following abbreviations are used in this manuscript: 674

CFD - Computational Fluid Dynamics	p_∞ - Infinite unperturbed pressure	691
LBM - Lattice Boltzmann Method	\hat{x} Unit vector in the x-direction	692
POD - Principal Orthogonal Decomposition	τ Viscous stress tensor	693
RSM - Response Surface Model	ν Kinematic viscosity	694
HF - High-fidelity (computation)	∇u - Velocity gradient tensor	695
AI - Artificial Intelligence	S_{eq} - Vehicle's surface	696
NN - Neural Network	\hat{e}_t Tangential unit vector	697
LLE - Locally Linear Embedding	S_{cp} - Cutting plane in the wake of the vehicle	698
CNN - Convolutional Neural Network	ρ Fluid density	699
SUV - Sport Utility Vehicle	U_∞ - Free stream velocity	700
$C_d A$ - Drag coefficient times frontal area	u_x - Velocity component in the x-direction	701
LES - Large Eddy Simulation	CAD - Computer-Aided Design	702
SGS - Subgrid Scale	\mathbf{p} Shape parameter vector	703
Re - Reynolds number	\mathcal{D} - Domain of admissible shape parameters	704
f_x - Drag force component in the x-direction		705
p - Static pressure		690

Appendix A. Mesh Matching 706

Appendix A.1. 707

If the initial geometry $\underline{S}(\cdot, \mu^{ref})$ and the deformed geometry $\underline{S}(\cdot, \mu)$ do not share the same mesh topology, as in the case of remeshing, establishing a correspondence between the two meshes is necessary. One approach to achieve this, is to employ optimal transport between the nodes or elements of the two meshes. In this context, the initial geometry $\underline{S}(\cdot, \mu^{ref})$ and the deformed one $\underline{S}(\cdot, \mu)$ are treated as point clouds of elements centroids, each consisting of N_v and N_v^μ points, respectively. The centroids are noted as $(c_j)_{j=1\dots M}$ for $\underline{S}(\cdot, \mu^{ref})$ and $(c_i^\mu)_{i=1\dots M^\mu}$ for $\underline{S}(\cdot, \mu)$, with each point assigned a mass m_j and m_i^μ , respectively. By default, point clouds are conceptualized as distributions of uniform mass where $m_j = \frac{1}{M}$ and $m_i^\mu = \frac{1}{M^\mu}$. 708
709
710
711
712
713
714
715
716

The transport plan π^μ between $\underline{S}(\cdot, \mu)$ and $\underline{S}(\cdot, \mu^{ref})$ is a matrix of dimensions $M^\mu \times M$, where m_{ij} represents the amount of mass transported from the initial point $c_i^\mu \in \underline{S}(\cdot, \mu)$ to $c_j \in \underline{S}(\cdot, \mu^{ref})$ (see Figure A1). The definition of the transport plan is formulated to minimize a transport cost, which is given by: 717
718
719
720

$$\text{cost}(\pi^\mu) = \sum_{ij} m_{ij} \|c_j - c_i^\mu\|^2 \quad (\text{A1})$$

Therefore each point $c_j \in \underline{S}(\cdot, \mu^{ref})$ is connected to one or multiple points $c_i^\mu \in \underline{S}(\cdot, \mu)$ with a specified weight m_{ij} if $m_{ij} > 0$. 721
722

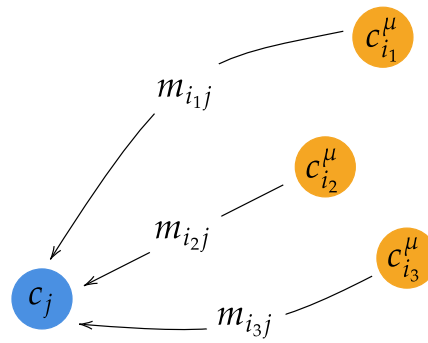


Figure A1. Transport plan towards a reference point c_j of the reference cloud. The m_{ij} represents the mass transport from point c_i^{μ} to point c_j

References

1. J.L. Lions; E. Magenes, Non-homogeneous boundary value problems and applications; Publisher: Springer-Verlag , 1972. 723
2. O. Pironneau, On optimum design in fluid mechanics. *Journal of Fluid Mechanics* **1974**, *64*, 97 – 110. 724
3. A. Jameson, Aerodynamic design via control theory. *J Sci Comput* **1988**, *3*, 233–260. 725
4. A. Jameson, Optimum aerodynamic design using CFD and control theory, AIAA paper 95-1729, AIAA 12th Computational Fluid Dynamics Conference, San Diego, CA, June 1995. 726
5. I. Cheylan ; G. Fritz ; D. Ricot ; P. Sagaut, Shape Optimization Using the Adjoint Lattice Boltzmann Method for Aerodynamic Applications. *AIAA Journal* **2019**, *57*, 1–16. 727
6. RL.Hardy, Multiquadric equations of topography and other irregular surfaces. *J. Geophys. Res.* **1971** , *76*(8), 1905–1915. 728
7. MD. Buhmann MD , *Radial Basis Functions: Theory and Implementation* , Publisher: Cambridge University Press, New York , 2003. 729
8. D. J. C. MacKay, Introduction to Gaussian processes , *NATO ASI series F computer and systems sciences* **1998**, *168*, 133–166. 730
9. K. Pearson, On Lines and Planes of Closest Fit to Systems of Points in Space , *Philosophical Magazine* **1901**, *2*(11), 559–572. 731
10. J. Lumley J ; *Stochastic tools in turbulence* ; . Publisher : Academic Press , New York 732
11. S.T. Roweis ; L.K. Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding , *Science* **2000**, 2323–2326. 733
12. S. L. Brunton; R. N Bernd ; P. Koumoutsakos, Machine learning for fluid mechanics, *Annual review of fluid mechanics* **2020** , *52*, 477–508. 734
13. K. Ando ; A. Takamura ; I. Saito, Automotive aerodynamic design exploration employing new optimization methodology based on CFD, *SAE Int. J. Passeng. Cars-Meek Syst* **2010** , *3*, 398–406 735
14. N. Umetani ; B. Bickel, Learning three-dimensional flow for interactive aerodynamic design, *ACM Transactions on Graphics* **2018**, *37*(4), 1–10. 736
15. A. Bertram ; C. Othmer ; R. Zimmermann, Towards Real-time Vehicle Aerodynamic Design via Multi-fidelity Data-driven Reduced Order Modeling, 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 737
16. E. Gunpinar ; U.C. Coskun ; M. Ozsipahi ; S. Gunpinar, A Generative Design and Drag Coefficient Prediction System for Sedan Car Side Silhouettes based on Computational Fluid Dynamics, *Computer-Aided Design* **2019**, *111*, 1–10. 738
17. F. Salmoiraghi ; A. Scardigli ; H. Telib, Free Form Deformation, mesh morphing and reduced order methods: enablers for efficient aerodynamic shape optimization, *International Journal of Computational Fluid Dynamics* **2018**, *32*, 1–10. 739
18. A. Badiás ; S. Curtit ; D. González ; I. Alfaro ; F. Chinesta ; E. Cueto, An augmented reality platform for interactive aerodynamic design and analysis, *International Journal for Numerical Methods in Engineering* **2019**, *120*(1), 125–138. 740
19. X. Li ; C. Xie ; Z. Sha, Part-Aware Product Design Agent Using Deep Generative Network and Local Linear Embedding, Hawaii International Conference on System Sciences, January 2021. 741
20. X. Guo ; W. Li ; F. Iorio, Convolutional Neural Networks for Steady Flow Approximation, In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining Association for Computing Machinery, New York, NY, USA, 481–490, 2016. 742
21. P. Baqué ; E. Remelli ; F. Fleuret ; P. Fua, Geodesic Convolutional Shape Optimization, 35th ICML 2018, Stockholm, Sweden , 481-490 . 743
22. Y. Afshar ; S. Bhatnagar ; S. Pan ; K. Duraisamy ; S. Kaushik, Prediction of aerodynamic flow fields using convolutional neural networks, *Computational Mechanics* **2019**, *64*, 525–545. 744
23. S. Jacob ; M. Mrosek ; C. Othmer ; H. Köstler, Deep Learning for Real-Time Aerodynamic Evaluations of Arbitrary Vehicle Shapes, *SAE Int. J. Passeng. Veh. Syst.* **2022** , *15*(2), 77–90. 745
24. S. Sun; Y.-P. Chang; Q. Fu; J. Zhao; L. Ma; S. Fan; B. Li; A. Shestopalov; P. Stewart; H. Friz, Aerodynamic shape optimization of an SUV in early development stage using a response surface method, *SAE International Journal of Passenger Cars-Mechanical Systems* **2014**, *7*, 1252–1263. 746
25. L.J. Clancy, Aerodynamics, Wiley, **1975**. 747
26. J. Anderson, Fundamentals of Aerodynamics, Sixth Edition, McGraw-Hill, **2016**. 748

27. P. Sagaut, Large Eddy Simulation for Incompressible Flows, Springer, **2006**. 767
28. M. Onorato ; A. Costelli ; A. Garrone, Drag Measurement Through Wake Analysis, SAE International Congress and Exposition, 85–93, **1984**. 768
29. J. Feydy ; T. Séjourné ; F.-X. Vialard ; S.-I. Amari ; A. Trounev ; G. Peyré, Interpolating between Optimal Transport and MMD using Sinkhorn Divergences, proc. of the *The 22nd International Conference on Artificial Intelligence and Statistics*, **2019**, 2681–2690. 769
30. M.G. Kendall, A new measure of rank correlation, *Biometrika*, **1938**, 30 , 81–93. 770
31. B. Song ; C. Yuan ; F. Permenter ; N. Arechiga ; F. Ahmed, Surrogate Modeling of Car Drag Coefficient with Depth and Normal Renderings, *arXiv*, **2023**. 771
32. E. Remelli ; A. Lukoianov ; S. R. Richter ; B. Guillard ; T. Bagautdinov ; P. Baque ; P. Fua, MeshSDF: Differentiable Iso-Surface Extraction, *arXiv*, **2020**. 772
33. R.B. Nelsen, Kendall tau metric, *Encyclopedia of Mathematics*, **1994**. 773
34. S. Xie ; R. Girshick ; P. Dollar ; Z. Tu ; K. He, Aggregated Residual Transformations for Deep Neural Networks, proc. of the *The 30th IEEE Conference on Computer Vision and Pattern Recognition*, **2017**, 5987–5995. 774
35. A. Heft ; T. Indinger ; N. Adams , Introduction of a New Realistic Generic Car Model for Aerodynamic Investigation, *SAE Technical Paper*, **2012**. 775
36. O. Ronneberger ; P. Fischer ; T. Brox,U-Net: Convolutional Networks for Biomedical Image Segmentation, *Medical Image Computing and Computer-Assisted Intervention*, **2015**. 776
37. N. Rosset ; G. Cordonnier ; R. Duvigneau ; Ad. Bousseau, UInteractive design of 2D car profiles with aerodynamic feedback, *Computer Graphics Forum*, **2023**. 777
38. F. Monti ; D. Boscaini ; J. Masci ;E. Rodola ; J. Svoboda ; M. Bronstein, M, Geometric Deep Learning on Graphs and Manifolds Using Mixture Model CNNs, *arXiv*, **2016**. 778
39. Nikita Durasov, Artem Lukoyanov, Jonathan Donier, and Pascal Fua. DEBOSH: Deep Bayesian Shape Optimization 779
39. N. Durasov ; A. Lukoyanov ; J. Donier ; P. Fua., DEBOSH: Deep Bayesian Shape Optimization, *arXiv*, **2021**. 780

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content. 790