Data-Centric Engineering (2026), 7: e29
doi:10.1017/dce.2026.10061

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Learning from data versus data for learning: a shape-to-aerodynamics study

Mark Benjamin and Gianluca Iaccarino

Department of Mechanical Engineering, Stanford University, USA
Corresponding author: Mark Benjamin; Email: markben@stanford.edu

Received: 19 November 2025 UTC; **Revised:** 23 March 2026 UTC; **Accepted:** 30 April 2026 UTC

Keywords: automotive aerodynamics; dataset characterization; diversity metrics; neural networks; shape to aerodynamics

Abstract

The cost of high-quality aerodynamics simulations for realistic automotive configurations makes comprehensive design studies unfeasible. Data-driven surrogates (learning from data) are an appealing alternative, and there is no shortage of approaches that target shape-to-aerodynamics predictions. However, there is a fundamental limitation (data insufficiency problem) in this context: owing to the proprietary nature of commercial automotive designs, training datasets are limited to a few freely-available geometries. In a previous work the authors, a strategy to construct datasets for training surrogates was introduced. It enables controlled generation of an arbitrary number of samples, by convex interpolation between a small number of basis geometries. In this work, we extend this strategy by introducing three features, namely size, density, and diversity that characterize more general datasets. These are important to assess how useful is a dataset for a specific prediction task (data for learning). A formal measure of diversity is developed and then, datasets of successively increasing diversity but constant size are constructed. We show that the dataset diversity has an impact on the predictive accuracy of machine learning surrogates. A power-law scaling, $\varepsilon \propto M^{1/2} m^{-1/6}$, where ε is the prediction error, M is the diversity, and m the dataset size, collapses 23 controlled experiments onto a single curve, revealing that diversity dominates size in determining prediction error. The proposed framework allows for more rigorous a priori evaluation of models than is currently possible and can be applied readily to other shape optimization problems.

Impact Statement

This work presents a novel characterization framework for datasets for learning shape optimization problems. It presents an advance in the quantification of dataset diversity that has implications for what data are collected and how the data are generated. While we apply the framework to an automotive aerodynamics problem, the method has broad applicability across engineering optimization.

1. Introduction

The exterior design of an automobile is a result of several considerations, including, among other things, safety, aesthetics, ergonomics, and function. An important concern is the reduction of aerodynamic drag, which, along with rolling resistance from the wheels, is one of two primary contributors to the energy

This research article was awarded Open Data badge for transparent practices. See the Data Availability Statement for details.

© The Author(s), 2026. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (http://creativecommons.org/licenses/by/4.0), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

consumption of the vehicle. Numerical prototyping is still an expensive way of design exploration for aerodynamic optimization, however. The complexity of modern vehicle designs leads to a high-dimensional parameter space that can be prohibitively expensive to sample. In addition, the evaluation of the quantity of interest (the drag coefficient) for each single design point is costly, requiring the solution of the Navier–Stokes equations at high Reynolds numbers, on a mesh sufficiently fine so as to resolve the details of configuration, with a model for either all or some of the turbulent scales. Surrogate models are therefore of interest; given the nonlinear nature of the governing equations and the complexity of the geometries of interest, data-driven models for drag prediction are a common choice. The primary issue with constructing surrogates in this application space is the absence of freely-available, realistic automotive geometries, owing primarily to the proprietary nature of commercial designs (see, for instance, the work by the Honda Motor Co. [Nagaoka et al., 2024], where CFD of several proprietary designs is validated against wind tunnel data).

The two factors that introduce complexity in this specific problem are: first, given the dimension and velocity typical of automotive cruise conditions, the flow regime involves a highly turbulent flow, with a wide range of scales generated. To a first order, assuming homogeneous, isotropic, and fully-developed turbulence, a typical passenger sedan cruising at 70 mph generates turbulent eddies of the order of 10 μm (Kolmogorov, 1941); this gives about six decades of size separation between the largest and smallest structures in the flow. This turbulence has to be accounted for, either through resolution of the relevant structures using grid and timestep resolution, or through modeling. The second factor is the complexity of automotive geometries (see Figure 1), which require a high surface mesh resolution to capture. Several models for turbulence have been developed for the commonly-used RANS (Reynolds-Averaged Navier–Stokes) approach (Wilcox, 2006), each with the goal of estimating the impact of the turbulent fluctuations on the time-averaged velocity and pressure fields, which are used to compute the aerodynamic quantities of interest.

One of the first automotive benchmark cases for CFD solvers was the so-called Ahmed Body (Ahmed et al., 1984). Over the years, several other configurations have been proposed as reference cases, with varying complexities and parameterizations. A detailed review of the early, simpler models is found in Le Good and Garry (2004). A more recent development is the DrivAer model (Heft et al., 2012a), a much

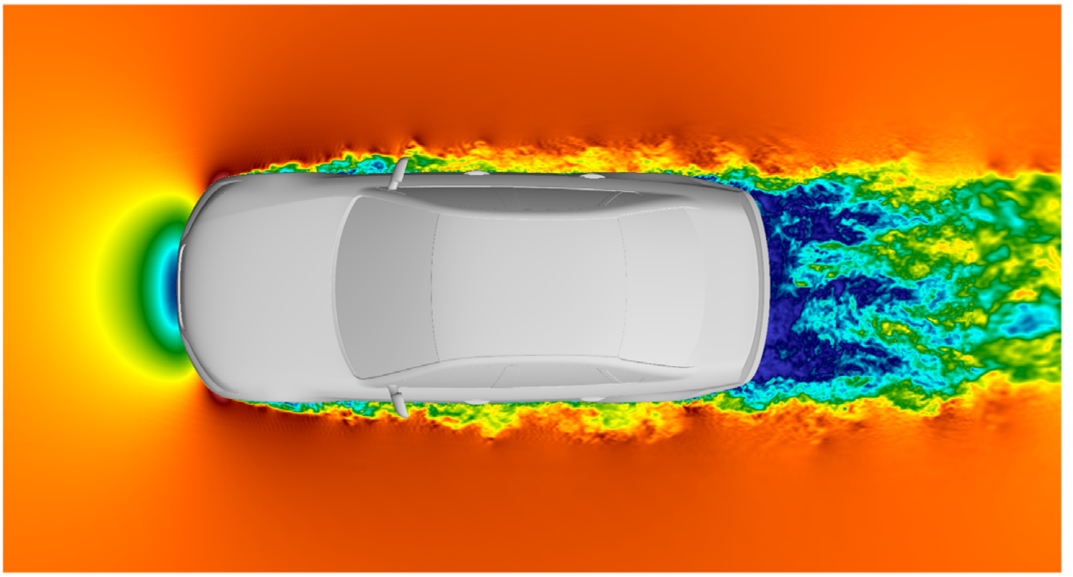


Figure 1. *Temporal snapshot of time-resolving simulation of flow past a car, showing velocity magnitude contours.* Image courtesy*: Kun Lu.*

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

more detailed and realistic model that has since been the focus of a number of workshops for the evaluation and validation of CFD solvers. With the DrivAer model, it has been found that RANS models struggle to predict the surface pressures accurately even at high resolutions of 300 million cells, and it is found that the scale-resolving Improved Delayed Detached-Eddy Simulations (IDDES) (Shur et al., 2008; Ashton et al., 2024) are much better than steady-state calculations (Ashton and Revell, 2015). With the advancements of computing power, particularly the rise in popularity of graphics processing units (GPUs) for code acceleration, turbulence modeling techniques that capture a portion of the turbulent scales on the grid are becoming more tractable, and it is seen that these methods provide superior accuracy to RANS simulations, particularly in flows with separation (Rodi, 1997).

1.1. Data for learning challenges

There have been attempts in the literature to address the issue of database generation in shape-to-aerodynamics problems. Viquerat and Hachem (2020) studied the problem of 2D airfoils in the laminar (Re 10) regime, and used a random process to generate Bézier curves with which airfoils were developed. Garcia-Fernandez et al. (2023) studied simplified representations of heavy vehicles, using CNNs to predict aerodynamic quantities of interest. Other parameterizations explored in the literature is the idea of using a PolyCube map to represent the geometry (Umetani and Bickel, 2018), or using parameterized curves to generated two-dimensional silhouettes of cars (Gunpinar et al., 2019), both of which require manual parameterization, and have been tested on relatively simple shapes. Song et al. (2023) used the versatile ShapeNet database, which is a large repository of freely-available CAD models organized hierarchically, in their investigations of automotive drag. The ShapeNet dataset (Chang et al., 2015) has been used in the vast majority of studies to provide training data for machine learning-based aerodynamics predictors (Umetani and Bickel, 2018; Rios et al., 2019, 2021; Remelli et al., 2020; Li et al., 2023; Song et al., 2023). The major limitation of ShapeNet is the lack of surface resolution in all of the models, so that the resulting model could not be considered a tool to aid even the initial stages of automotive aerodynamic design.

An approach taken in the literature has been to develop a database with manual parameterizations of the much higher quality DrivAer model discussed earlier (Elrefaie et al., 2024a, 2024b). For example, Elrefaie et al. (2024a) generate a database using 29 parameters identified across the entire DrivAer model. The drawbacks to this approach are twofold: first, manual parameterization is labor and time-intensive; second: there is no guarantee that the resulting database is an exhaustive representation of relevant shapes. This approach is thus not scalable beyond limited scope, such as, for example, predicting drag in minor variations of a specific model of automobile.

In this work, we build on the approach introduced by the authors Benjamin and Iaccarino (2025c), henceforth referred to as ShapeCH. Using this approach, it is possible to build a controlled set of database samples using a small number of basis shapes. Using databases generated using this approach, we will address what we term the dataset characterization problem, which can be summarized as follows: there is at present no method to formally make comparisons between different datasets a priori. Table 1 shows a list of currently available datasets for automotive aerodynamics predictions. The majority of databases are derived from either the ShapeNet or DrivAer models, with others using even simpler geometries. However, since these database do not come with a formal characterization of diversity or variety, it is not possible to make any assessments about which database might be suitable for a given application: say, for instance, making predictions on hatchbacks alone. In this case, one might intuitively expect a dataset containing many representative samples of hatchbacks to perform better than one not containing any, but at present there is no way of characterizing this information. Thus, the only way to evaluate different datasets is to train models on them and make predictions, comparing them with any reference drag data that might be available. It is evident that this is not a feasible proceeding in most situations. In the complex realm of automotive design, there is a lack of clarity on the additional coverage in the feature space provided when a new geometry is considered, owing to the nominally high-dimensional parameter space

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

Table 1. *Datasets for data-driven automotive aerodynamics surrogates*

Dataset	Size	Based on
Umetani and Bickel (2018)	889	ShapeNet
Baque et al. (2018)	2,000	–
Gunpinar et al. (2019)	1,000	2D silhouettes
Remelli et al. (2020)	1,400	ShapeNet
Usama et al. (2021)	500	2D silhouettes
Jacob et al. (2022)	1,000	DrivAer
Rios et al. (2021)	600	–
Li et al. (2023)	551	Ahmed body
Li et al. (2023)	611	ShapeNet
Song et al. (2023)	2,474	ShapeNet
Elrefaie et al. (2024a)	4,000	DrivAer
Trinh et al. (2024)	1,121	–
Elrefaie et al. (2024b)	8,000	DrivAer
Qiu et al. (2025)	12,000	DrivAer

that the designs live in. This also leads to confusion over whether predictions made by models trained on these datasets are actually interpolations or extrapolations.

1.2. Learning automotive aerodynamics from data

Several approaches have been taken in the literature to develop neural network-based models for automotive aerodynamics.

Planar projections of the automobile can be used, these views can be flattened by applying developable transformations to them (Benjamin et al., 2023); these representations can be parsed by CNNs (Song et al., 2023). The images can contain information about depth of field, as well as surface normals. The two main drawbacks of this approach are the forced uniform resolution of the pixelization over all regions of the image (which has been somewhat, but not completely solved by feature pyramid approaches [Lin et al., 2017]) and the problem of occlusion in nonconvex shapes. The latter problem is application specific; occlusion is a more significant issue when the geometry of interest has several internal structures (such as the internal cooling passages in a turbomachinery blade) than in exterior automotive surfaces, for examples. A possible solution is to utilize several slices through the geometry, rather than just one. An extension of CNNs to three dimensions, with 3D convolutions operating on voxel grids, is possible but not desirable due to computing and storage requirements scaling cubically with the characteristic length scale resolved by the voxel (Maturana and Scherer, 2015).

Another common representation of a surface is a point cloud, which is simply an unordered set of vectors that contain coordinate information. The format is simple and low storage compared to three-dimensional voxelizations, and PointNets have been developed as an architecture class uniquely suited to working with point clouds (Qi et al., 2017a). The key to working with point clouds is to ensure invariance of the functions in the network to permutations of the vectors in the set, and the PointNet architecture achieves this using a symmetric max pooling function. Subsequent iterations of the architecture (Qi et al., 2017b) use hierarchical feature extraction to compensate for the lack of connectivity information in the point clouds. Owing to their innate hierarchical construction, CNNs tend to generalize better than PointNets. Surface tessellations or meshes are a popular representation choice, and MeshCNNs (Hanocka et al., 2019) operate directly on meshes, using edges as the basis element to apply convolutions. MeshCNNs are tailored to work with triangular tessellations, and see applications in segmentation tasks for 3D models. For a more flexible approach, the connectivity information in meshes can be assembled into graphs. Graph neural networks (GNNs) are designed to take advantage of this format (Wu et al., 2021),

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

using message passing to aggregate information from neighboring nodes. A development that enables the use of convolutions are the Graph Convolutional Networks (GCNs) (Kipf and Welling, 2017). While GCNs handle spatial locality more naturally than PointNets, they can be difficult to train. In addition, they are much more sensitive to perturbations to the graph connectivity than PointNets. Scalability is also a concern with GCNs for large graphs. Combinations of the earlier architecture types have been developed more recently, such as the Point–Voxel CNN architecture (Liu et al., 2019), which combines the benefits of PointNets and VoxelNets, and Regularized Graph Convolutional Neural Networks (RGCNNs) (Te et al., 2018), which are a combination of point cloud representations with graph neural networks. Another recent development is the Geometry-Informed Neural Operator (GINO) (Li et al., 2023), which is a hybrid architecture that combines GNNs and Fourier Neural Operators (FNOs) (Li et al., 2020). A summary of the earlier representation types and the corresponding network architectures is shown in Table 2.

The database generation procedure introduced in this work has the benefit of being compatible with any of the network architectures listed ealier, and the choice of specific representations and corresponding network types is made with a view to make predictions with accuracy suitable for the datasets under consideration, not to present a comparative analysis of different architectures. The focus of the work is solely on the dataset construction.

In a previous work by Benjamin and Iaccarino (2025c), a methodology is developed for generating geometries for training in a controlled manner, starting with a small number of starting cases. Using the database developed in that work, we develop a measure of diversity that can be used to characterize different datasets.

2. Database generation, simulation, and model training

2.1. Database generation using ShapeCH

ShapeCH is a procedure that uses a limited number of starting (or basis) geometries that form a convex hull (CH) in the space of 3D shapes. From these basis configurations, intermediate geometries are developed by shape interpolation of the corresponding Signed Distance Function (SDF). The methodology is described in detail in Benjamin and Iaccarino (2025c), but we provide a succinct summary here: first, the basis STL tessellations are converted into SDFs using structured ray-tracing along orthogonal axes. Rays cast from bounding box faces detect surface intersections via the Moller–Trumbore algorithm, creating binary voxel grids (BVGs). The distance transform of the BVGs yields continuous SDF representations where positive values indicate interior regions and negative values exterior space. Then, barycentric interpolation is performed within the convex hull of the basis geometries. Given n input SDFs ϕ_i^p, interpolated shapes are computed as $\overline{\phi}_i(d) = \sum_{p=1}^{n} w_p \phi_i^p(d)$ subject to $\sum w_p = 1$ and $w_p \geq 0$. This weighted combination preserves topological features while enabling smooth morphing between basis configurations. Finally, zero-level sets are extracted from interpolated SDFs and triangulated surfaces are reconstructed using the marching cubes algorithm. In addition to the automotive aerodynamics application in the original work by Benjamin and Iaccarino (2025c), the work has been used in generating datasets for wind loading predictions on buildings (Vargiemezis et al., 2025).

Table 2. *Surface representations and corresponding network types*

Surface representation	Network type
2D images	CNNs
3D binary voxel grids	VoxNets
Point clouds	PointNets
Triangular meshes	MeshCNNs
Graphs	GNNs
Signed distance functions (SDFs)	GINOs

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

2.2. Geometry and the DrivAerCH3 database

To best represent the complexities of a realistic automobile, we consider the DrivAer model discussed earlier, an open-source midsize passenger car geometry developed to assess the quality and contrast automotive aerodynamics investigations carried out using computational fluid dynamics tools (Aultman et al., 2022). The model is available in three configurations: a fastback, estate, and sedan, shown in Figure 2.

We construct the dataset by using the three back configurations of the DrivAer model as the basis elements of the convex hull (Benjamin and Iaccarino, 2025c). The identifier applied to this dataset is "DrivAerCH3," where the "CH3" represents that the convex hull is generated from three basis geometries (a triangle in 2D). In general, with n basis geometries we construct a simplex in $n-1$ dimensions. Note that no other user-specified input is required to generate infinite *new* configurations by sampling in the SDF space. The three configurations are different enough so as to provide a sufficient variation in drag coefficient (the separation aft of the vehicle being significantly impacted by the form of the rear roof slope), but also similar enough to provide a controlled testing ground for the proposed methodology.

2.3. From a database to a dataset: large eddy simulations

Large eddy simulations (LES) are used to obtain the ground truth aerodynamic quantities of interest for the training dataset. Large eddy simulations directly resolve the largest, energy-containing scales of turbulent flows, and model the smallest scales. This is achieved by the application of a low-pass filter to the governing Navier–Stokes equations of motion, whereby scales smaller than the filter are removed. In common practice, high-frequency modes are removed implicitly by a combination of the finite sizing of the mesh and the filtering effect of numerical discretization of the equations.

LES simulations are carried out at fixed speed (85 mph) and without wheel rotation, using the CharLES solver. Details of the Voronoi diagram-based meshing paradigm used in the tool can be found in Brès et al. (2018), and the formulation of the low-Mach Helmholtz pressure solver is outlined in Ambo et al. (2020). The code has been validated in aerodynamics analysis of a number of high-Reynolds number flow studies (Goc et al., 2021). The current simulations use ~2 million control volumes, with refinement near the surfaces of the car as shown in Figure 3. The mesh resolution on the surface is $\Delta/L = 0.0023$, where Δ is the local mesh size and L is the characteristic length of the car. The domain is sized at $15L \times 8L \times 6L$, with the car placed at $5L$ from the inlet. Since the Helmholtz formulation permits low-frequency pressure oscillations, we use a numerical sponge at $2L$ from the domain boundaries to reduce the time for the reflecting waves to be damped out. The Vreman subgrid model is used to approximate the effects of the subgrid scales on the resolved scales, and walls are modeled using the algebraic wall model. The flow is initialized with a converged initial condition from the baseline DrivAer notchback geometry. The simulation is run for a total time of 2 seconds, and statistics are collected for the last 1 second. The simulation parameters are summarized in Table 3. We choose to limit our experiments to static wheels for the following two reasons: first, while wheel rotation is important to capture absolute count variations in drag, and important in industrial settings, we are simply concerned with a representative dataset where drag "delta" (the difference in drag between different simulations) is represented with sufficient accuracy



Figure 2. *DrivAer back configurations: estateback (left), fastback (center), and notchback (right).*

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

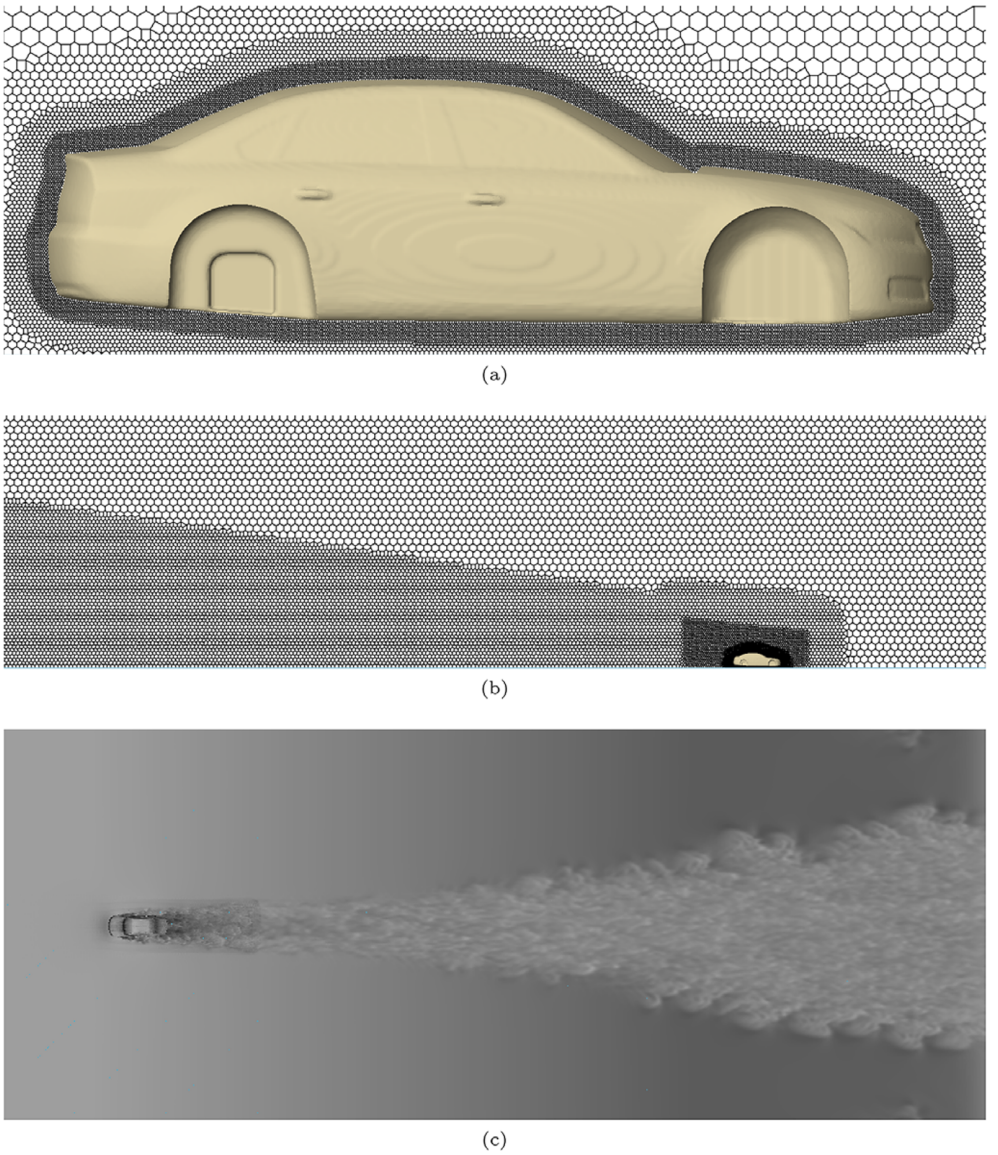


Figure 3. *View of the mesh used for the large eddy simulations, with refinement zones shown: (a) near the car, (b) in the wake. (c) Instantaneous velocity magnitude contours showing wake behind the car.*

so as to demonstrate our dataset generation and characterization framework; second, it has been shown in Nagaoka et al. (2024) that moving mesh simulations of wheels provide significant accuracy gains for automotive simulations compared to other methods such as rotating boundary conditions, chiefly owing to the non-axisymmetric nature of the wheel rims. However, the moving mesh simulations were too expensive for the computational budget of this study, due to the large number of geometries simulated. Hence, we confine our attention to simulations with no wheel rotations.

The main output of interest for this study is the drag coefficient, computed as $C_d = 2F_d/\rho v^2 A$, where F_d is the drag force, ρ is the fluid density, v is the freestream flow velocity, and A is the frontal area of the car projected in the direction of the flow.

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

Table 3. *Simulation setup details for the DrivAerCH3 database*

Domain size	$15L \times 8L \times 6L$ (streamwise, spanwise, wall-normal)
Mesh size	2 MCVs
Wall model	Algebraic
Subgrid model	Vreman
Simulated time	2 s
Collected data	Averaged drag coefficient, pressure coefficients
Total number of simulations	1,275

The wall time for a single simulation is about 12 minutes on a NVIDIA H100 GPU (Choquette, 2023). We make comparisons with the original set of experiments carried out by Heft et al. (2012b). The configuration that matches the one that we use is the Fastback, with the smooth underbody, without mirrors, with wheels (the *F_S_woM_wW* experiment from Figure 9 of Heft et al.). The reported C_D in that case is 0.227, which compares well with our value of 0.223. Our LES predictions match the experiments better than the Heft et al. RANS simulations which give a value of 0.233. We construct a refined grid of 4M CVs and verify that the difference between the two grids is less than 1% at the pixel resolution used for the CNNs (see Figure 4). We note that the experiments in the article were at a lower Reynolds number than ours ($7M$ vs. $10M$), however, the C_D is observed to plateau around that Reynolds number, so we do not believe this has a significant impact. The selected Reynolds number is chosen to correspond with typical highway speed limits.

2.4. Model architecture

We use the image-based representation and convolutional neural network model (U-net) architectures found in Benjamin and Iaccarino (2025c). The inputs to the model are three planar projects of the geometry: the back, the side, and the top (shown in Figure 5), stacked to make three channels. The image resolution is 384×384, which is chosen to match the resolution of the mesh cells of the CFD and the resolution of the ray tracing. This choice is done in keeping with the need to resolve sufficiently fine changes to the surface. Concretely, taking the different discretization choices (ray tracing, mesh, and image resolution) into account, a geometry modification of about 12 mm to a surface can be detected in a vehicle that is 4 m long. The data in the images are binned into 255 bits.

Figure 6 shows a selection of samples from the dataset. One can see how the roofline is gradually morphed to produce designs intermediate to the basis cases. It is also evident how regions of low pressure

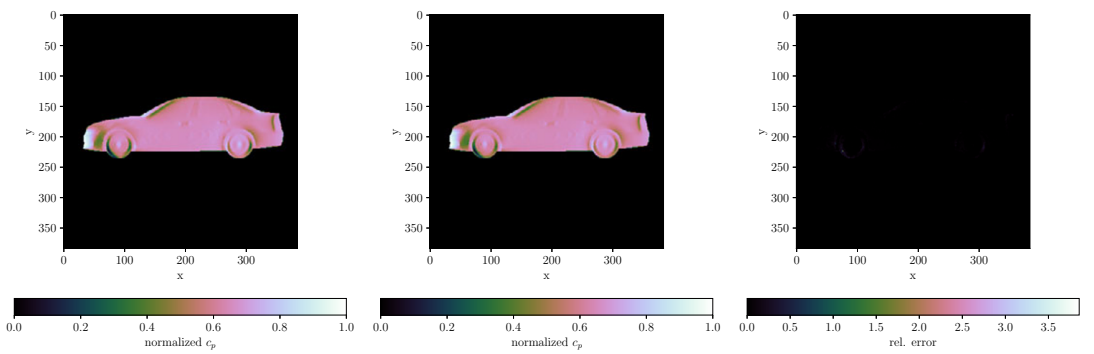


Figure 4. *Planar images of the left view of the notchback, showing (left) the normalized surface pressure coefficient computed on the 2M CV grid, (center) the normalized surface pressure coefficient computed on the 4M CV grid, and (right) the relative error between the two solutions.*

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

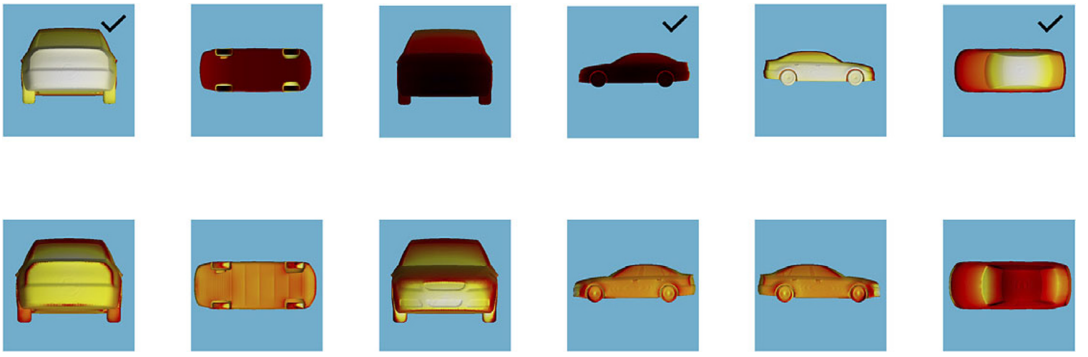


Figure 5. *Planar images of the normal distance from the camera (top row) and the normalized surface pressure coefficients (bottom rows) for a sample DrivAer geometry. The input images with the check marks (and the corresponding output views) are used for training.*

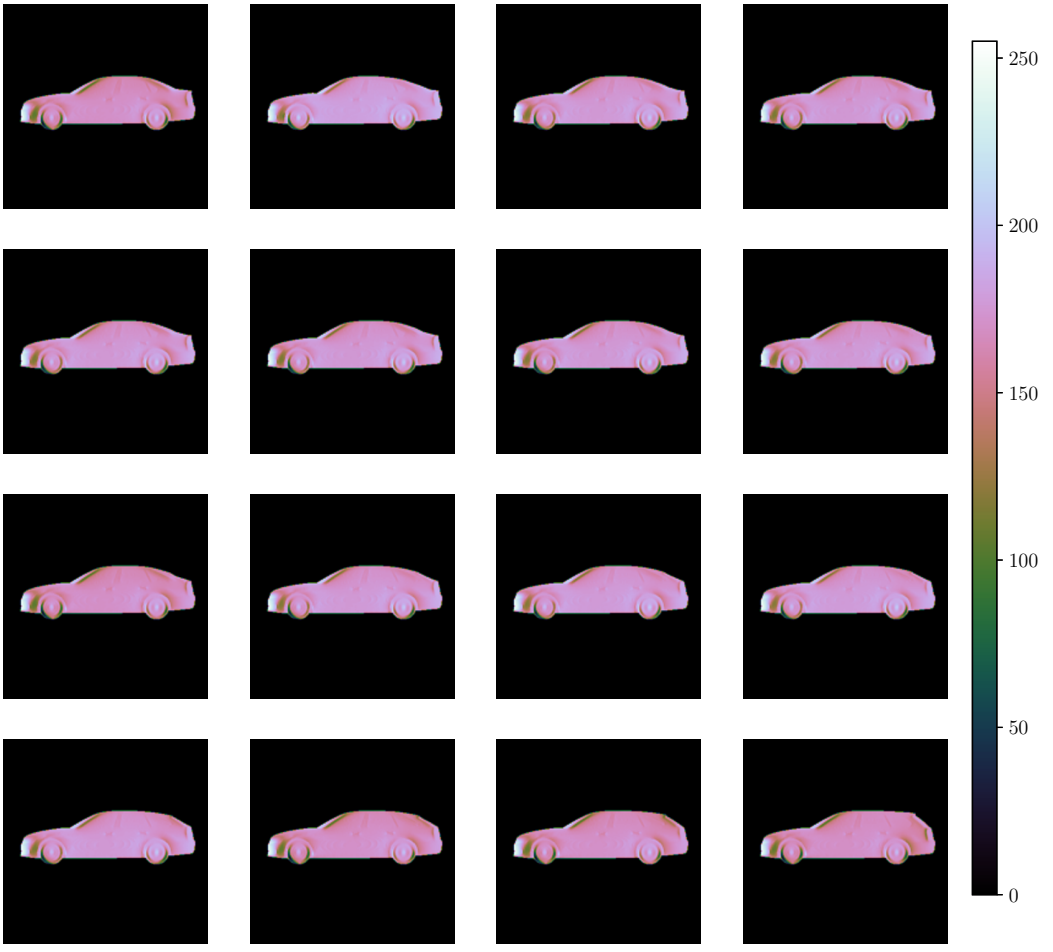


Figure 6. *Surface pressure coefficients (c_p) shown on left side views of a selection of samples in dataset, showing a transition from the notchback (top left) to the estateback (bottom right), with the fastback visited along the way. Values are normalized between 0 and 255 per sample.*

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

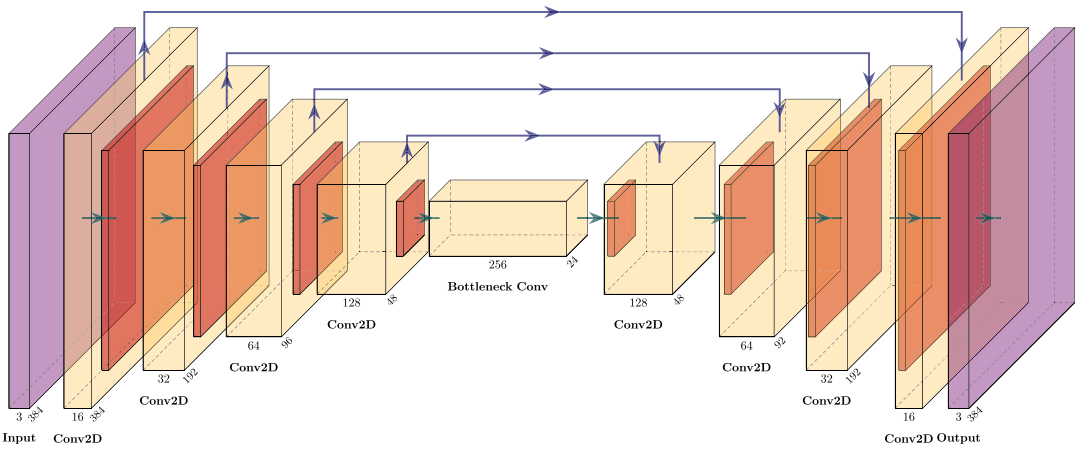


Figure 7. *Schematic of U-Net architecture for the c_p model.*

Table 4. *Network architecture*

Architecture	CNN/U-Net
Activations	ReLU
Pooling	Max
Loss	MSE
Epochs	500
Optimizer	Adam
Learning rate	10^{-3}
No. of parameters	50K (1D model), 250K (2D), 1M (3D)

are more apparent in the more estate-like designs, which is a result of the larger separation bubble that design tends to produce. With this method of generating training samples, the parameterization is not tied to discrete design features such as the door mirrors or handles (Benjamin et al., 2023), but a comprehensive three-dimensional morphological operation, designed to provide a high density of data point clustering in feature space.

The U-Net downsamples the input (dimension $384 \times 384 \times 3$) to a bottleneck of $24 \times 24 \times 256$ (Figure 7). The hyperparameters (learning rate, filters per layer, and optimizer type) are tuned by grid search. Architecture details are in Table 4. For a full review of the results of the DrivAerCH3 model, we refer the reader to the ShapeCH paper (Benjamin and Iaccarino, 2025c).

3. Dataset characterization

With the background for the datasets in place, we now focus on the topic of dataset diversity. This is of critical importance in the current application, given the extremely high number of parameters requires to describe typical automotive shapes. The question we address is simply this: given two datasets A and B of identical size, how would one determine which dataset is better? Traditionally, attributes used to describe datasets have an emphasis on the quality of individual data points, providing quantification of statistical bias, correlation, or experimental noise (Brazdil et al., 2022). However, if all the datasets under consideration are from a single, sufficiently reliable origin, such metrics are likely to be fairly uniform across all of them. The goal of dataset characterization under such conditions is to provide a measure of how well the underlying problem domain (i.e., the space of possible geometrical configurations) is

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

represented. If one seeks a universal predictor of aerodynamic drag for automobiles under given flow conditions, dataset characterization should assist in the selection of the dataset that provides most coverage of the feature space. Assuming that factors such as the dataset size, data quality, distribution of target values, and variability in input features are the same, one would like to distinguish between a dataset with only minor variations of a certain type of vehicle, and one with many different vehicle types represented. It can be argued that with access to an infinitely large database, it would be possible to achieve a true universal prediction, but in practical terms there is either scarcity of data or cost constraints that require a decision of how to select data for learning. In the following, we formalize these notions and ultimately derive an empirical scaling law that quantifies how prediction error depends on both diversity and dataset size. We propose to characterize dataset using three features:

1. *Size:* The number of samples in the dataset. A larger size is correlated with better generalizability and reduced overfitting of the model to the training data. Deep networks empirically perform better when trained on datasets of large size (LeCun et al., 2015).
2. *Density:* The number of samples per region of feature space. In $\mathbb{R}^1$, this reduces to the number of samples per unit length. A higher concentration of samples improves interpolative capabilities. The concept of density is aligned closely to the idea of *granularity* in computer vision applications (Cui et al., 2019). In general, in the absence of control over the data-sampling process, density can be different in different regions of the input space.
3. *Diversity:* A measure of the variety of the feature space, reducing to the span in $\mathbb{R}^1$. Higher diversity can aid with generalization by exposing the model to a broader range of inputs, reducing bias toward specific subsets of the inputs.

An illustrative example in $\mathbb{R}^1$ is shown in Figure 8, for training data collected from an underlying sinusoidal signal. It is seen that two of the three characteristics change simultaneously whenever a change to the dataset is made. This example also shows how diversity can be thought of as improving the representation of the underlying truth in the training set.

ShapeCH allows for a natural definition of the earlier characteristics. The size of the dataset is trivially defined as the total number of samples drawn in the convex hull. The density is defined as the number of samples per dimension of the simplex (see Figure 9). There are alternate ways to define this quantity: for instance, one could use the inverse of the average volume of the Voronoi cell containing the sample. This

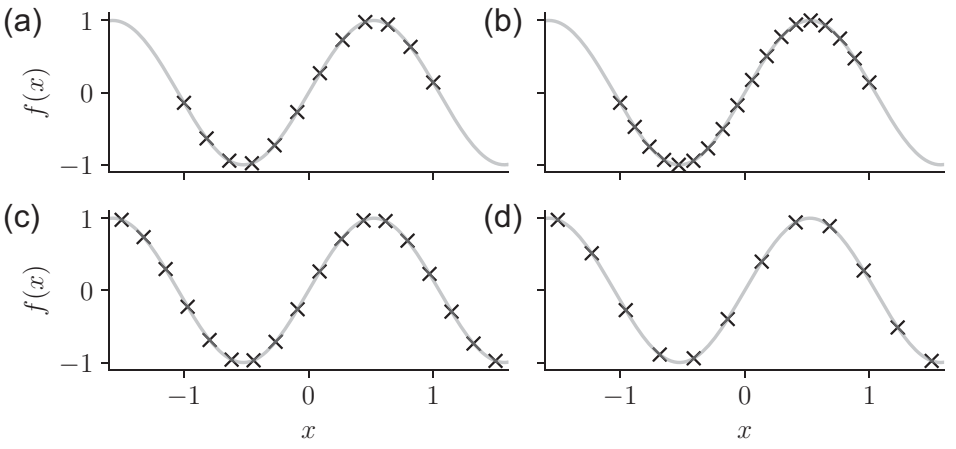


Figure 8. *Illustration of size, density, and diversity of a dataset in $\mathbb{R}^1$. Crosses represent training samples. Gray solid line represents the underlying truth. (a) Baseline; (b) same diversity, but different size and density; (c) same density, but different size and diversity; (d) same size, but different diversity and density.*

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

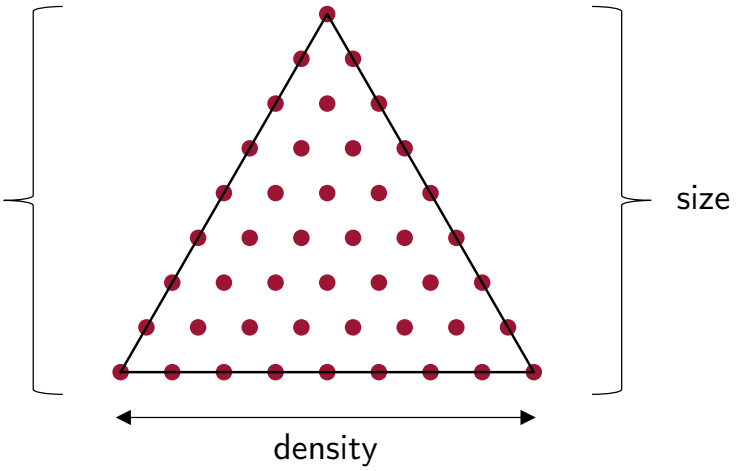


Figure 9. *Illustration of database size and density in the ShapeCH framework.*

could be a subject of future study, particularly in the context of nonuniform sampling. Additional nuance may be added to this definition when more complex forms of high-dimensional sampling are used. The diversity of the dataset is correlated to two disparate effects: the dimensionality of the simplex, and the variety in the basis cases. As an illustration, one might generate a more diverse training set by moving from a p- to a $(p+1)$-dimensional simplex by adding a new basis case. One could also affect the diversity by staying with a p-dimensional simplex, but replacing one of the basis cases with a geometry that is more "different" by some measure than the existing case, leading to more variety in the interpolated shapes.

3.1. A metric for diversity

In devising a metric for diversity, the following properties are desirable:

- The metric should be a non-negative scalar for a given dataset.
- The metric should be higher for more diverse datasets.
- The metric should be independent of the units or scale of the problem (i.e., invariant under uniform scaling of geometry and domain).
- The metric should be independent of the dimension of the simplex.
- The metric should be sensitive to affine dependence in the basis cases; that is, if one of the corners of the barycentric triangle was obtained by taking an affine combination of the other corners, the diversity metric should reduce. A subset of this condition is that if one of the corners is directly replaced with one of the others, the diversity metric should reduce as well.

It is evident that STL representations of geometries or surface meshes for simulations are not easy to work with when it comes to making assessments of differences between two cases, because the unstructured tessellation is not straightforward to standardize. However, the SDFs are an ideal representation with which to make point-to-point comparisons. Thus, given the above requirements, the following metric is proposed: given n discrete SDFs (i.e., basis configurations), $\phi_1, \phi_2, \ldots, \phi_n \in \mathbb{R}^{N_x \times N_y \times N_z}$, the pairwise L-2 distance between two SDFs ϕ_p and ϕ_q is computed as

$$d_{L2}\left(\phi_p, \phi_q\right) = \left(\sum_{x_i} \left(\phi_p(x_i) - \phi_q(x_i) \right)^2 \right)^{1/2}.$$

The pairwise distance matrix D is constructed where

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

$$D_{pq} = \frac{d_{L2}\left(\phi_p, \phi_q\right)}{\sqrt{N}}, \tag{1}$$

where N is the total number of points in the voxel grid.

To obtain a scalar diversity metric, the off-diagonal elements of the distance matrix are averaged:

$$M_0 = \frac{1}{n(n-1)} \sum_{p \neq q} D_{pq},$$

where n is the number of basis SDFs (i.e., $n = p + 1$, with p the simplex dimension). To ensure that the metric is scale independent (i.e., invariant to uniform scaling of the geometry and domain), the final diversity metric is normalized by the box length L_{box}:

$$M = \frac{M_0}{L_{\text{box}}}.$$

Combining the earlier steps,

$$M = \frac{1}{L_{\text{box}}} \cdot \frac{1}{n(n-1)} \sum_{p \neq q} \frac{d_{L2}\left(\phi_p, \phi_q\right)}{\sqrt{N}}. \tag{2}$$

It is noted here that the metric is a comparative one; the numerical value obtained from a dataset only makes sense when compared against the value obtained from another dataset; the one with the higher value has higher diversity. The development of diversity metrics of this kind is seen in the functional ecology literature (Mammola et al., 2021) to enable quantification of the extent of trait differences across different species of plant or animal. It is important to emphasize that the diversity of the databases generated using ShapeCH depends *only* on the basis cases.

To validate the metric, it is first applied to simple geometric databases.

As a simple model problem, consider the database generated from the convex hull of a unit sphere, a unit cube, and a unit right cone. The centroids of the shapes are aligned, and their SDFs obtained by ray tracing at a resolution of 32^3. The diversity metric M is computed for a variety of configurations, and the values shown in Table 5. A baseline value of 1.83×10^{-6} is obtained for the value of M. When the right cone is replaced by an ellipsoid with major and minor axes half-lengths of 1 and 0.5, respectively, M drops to 1.77×10^{-6}, which is consistent with our expectation that the convex hull of these three shapes is smaller, and therefore less diverse, than the convex hull including the cone. Likewise, when there is duplication of the sphere in the basis set, we see that the metric falls even further: a duplicated element lies at the centroid of the pair, minimizing its contribution to diversity. And finally, a basis set containing the same sphere repeated returns a diversity metric of 0, as expected.

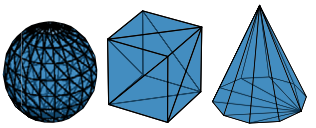
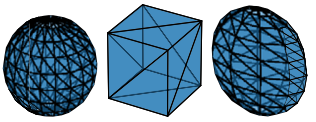
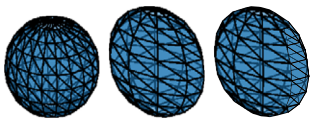
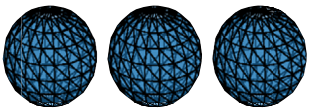
As a second validation, consider three spheres with tessellations of varying refinement: the first has 2,400 triangles, the second 32,000, and the third 240,000. The three shapes nominally represent the same object, but are subject to a small surface discretization error due to faceting. For a dataset generated with these bases, we expect the diversity metric to be small, yet nonzero. Accordingly, the metric is computed to be 1.3×10^{-9}, which is very small. The value is computed for a ray-tracing resolution of 256^3 rays, and changes slightly to 1.34×10^{-9} for 512^3 rays, due to an interaction of the approximations introduced by the tessellation and the ray tracing.

3.2. A hierarchy of datasets

To demonstrate the effects of varying diversities and to prepare the ground for a quantitative scaling law relating diversity to prediction error, a hierarchy of datasets is constructed, with an increasing number of basis cases. The number of samples is kept approximately constant. Thus, in the higher dimensional datasets, the diversity increases, while the density decreases, as the number of points per dimension of the

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

Table 5. *Diversity metric for model databases*

Database basis shapes	M
	1.83×10^{-6}
	1.77×10^{-6}
	8.03×10^{-7}
	0

Note. All SDFs are computed at 32^3 resolution.

simplex decreases. A total of six basis cases is considered: the first three are the variations of the DrivAer model from the previous sections, the next three are configurations of the AeroSUV model (Zhang et al., 2019), developed to be an SUV (Sports Utility Vehicle) equivalent to the DrivAer. Figure 10 shows how the three SUV models compare against the sedan. The wheelbase of both the SUV and sedan being identical ensures that the intermediate geometries have the same wheel assemblies; in general, this would not be true for arbitrary basis cases. However, the method is not conditioned on the interpolated shapes having a certain degree of verisimilitude; any shapes generated will still be used to "teach" the neural network about the drag response, providing a useful data point.

The basis cases that make up each dataset are detailed next. Each dataset comprised approximately 750 samples; for each dataset, the density (or number of samples per dimension of the simplex) is chosen to be the value that results in the dataset size being closest to 750 for a uniform, full-factorial sampling. For convenience, each dataset is named after the polygon that has a number of sides corresponding to the number of basis cases.

1. *The* DrivAerCH3 *dataset:* This dataset is generated from a 2D simplex using the DrivAer estateback, fastback, and notchback as basis cases.
2. *The* DrivAerCH4 *dataset:* This dataset is generated from a 3D simplex using the DrivAer estateback, fastback, notchback, and AeroSUV estateback as basis cases.
3. *The* DrivAerCH5 *dataset:* This dataset is generated from a 4D simplex using the DrivAer estateback, fastback, notchback, AeroSUV estateback, and fastback as basis cases.
4. *The* DrivAerCH6 *dataset:* This dataset is generated from a 5D simplex using the DrivAer estateback, fastback, notchback, AeroSUV estateback, fastback, and notchback as basis cases.

The details of the earlier datasets can be found in Table 6. With the samples generated, LES is performed of all cases under identical conditions as discussed earlier, taking 1330 H100 core-hours in total. As one

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

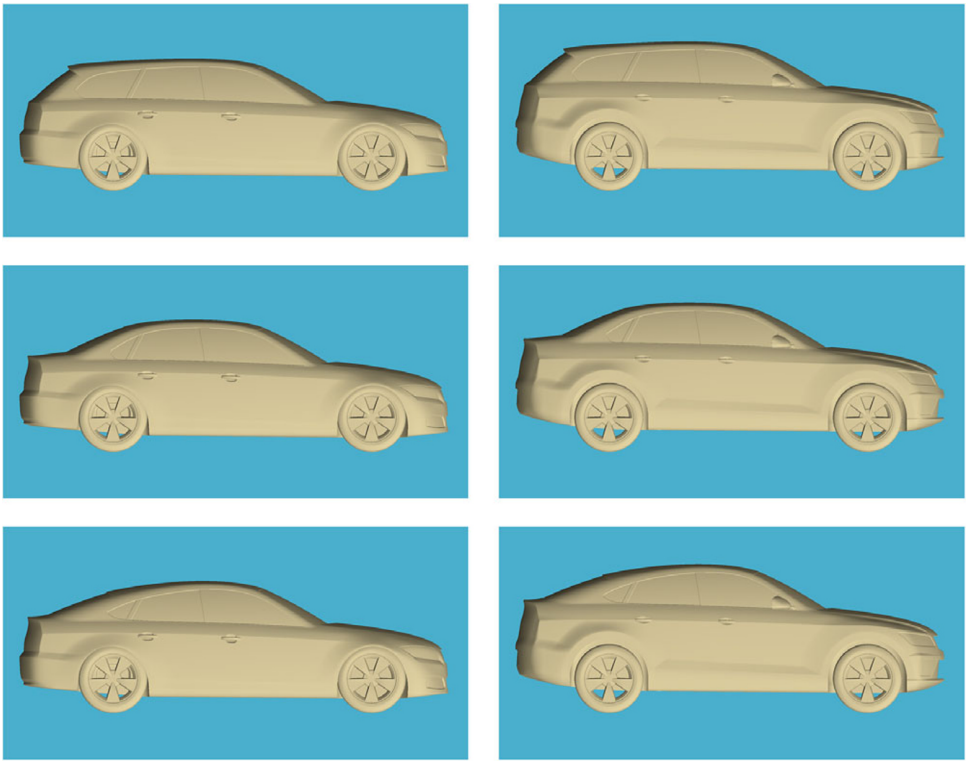


Figure 10. *The six basis STLs used in generating the datasets. Left column: DrivAer models; right column: equivalent AeroSUV models.*

might expect, there is a significant increase in the range of the drag coefficient with the addition of the SUV model. Figure 11 shows the distribution of C_D values for each of the four datasets. The first dataset, which has the same diversity as the original dataset (but a lower density and size), has a much narrower range of values, and the diversity in the output space increases as the diversity of the dataset increases.

While DrivAerCH3 provides us with a convenient barycentric triangle, the higher dimensional simplices of the DrivAerCH4 through the DrivAerCH6 datasets are more difficult to visualize. As an aid to dataset interpretation, we employ the multidimensional scaling (MDS) (Cox and Cox, 2000) technique to project the datasets down to two dimensions. MDS preserves the distances between different samples (in these cases, the distances are given naturally by the interpolation weights), in the lower dimensional space. Figures 12–15 show the MDS projections of the DrivAerCH3, DrivAerCH4, DrivAerCH5, and DrivAerCH6 datasets, respectively. Each sample is colored by the drag coefficient obtained from the LES. The drag coefficient is not used in defining similarity between samples; the distances are based on the feature space.

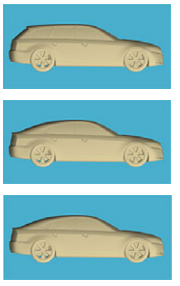
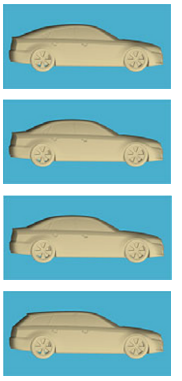
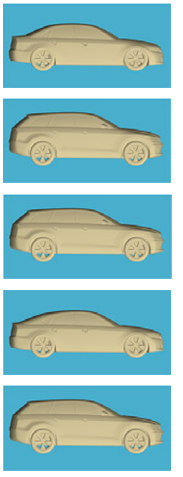
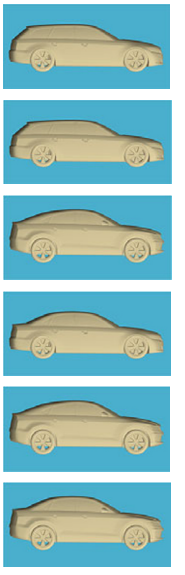
4. Results

4.1. Network performance

As the diversity of a training dataset grows (and its density decreases), given a fixed size, one expects to see an impact on the performance of models trained on DrivAerCH3 versus DrivAerCH6. The performance of the models in predicting the drag coefficients is shown in Figure 16. As the diversity of the datasets increases, the accuracy drops; more data are required to learn the increased variety in the training set. The reduced density of sampling leads to degraded interpolative performance.

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

Table 6. *Datasets' characteristics*

	DrivAerCH3	DrivAerCH4	DrivAerCH5	DrivAerCH6
Basis cases				
Dimension p	2	3	4	5
Size m	820	816	715	792
Density g	39	15	9	7
Diversity M	5.253×10^{-8}	1.047×10^{-7}	1.067×10^{-7}	1.012×10^{-7}
Training samples	650	650	650	650
Validation samples	120	116	15	92
Testing samples	50	50	50	50

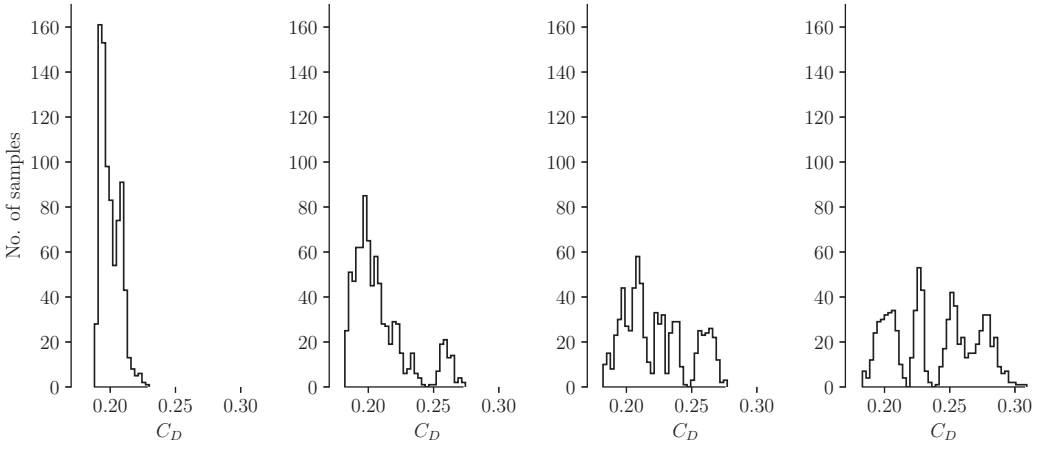


Figure 11. *Histograms of drag coefficients of different datasets. Left to right: the DrivAerCH3 dataset, the DrivAerCH4 dataset, the DrivAerCH5 dataset, and the DrivAerCH6 dataset.*

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

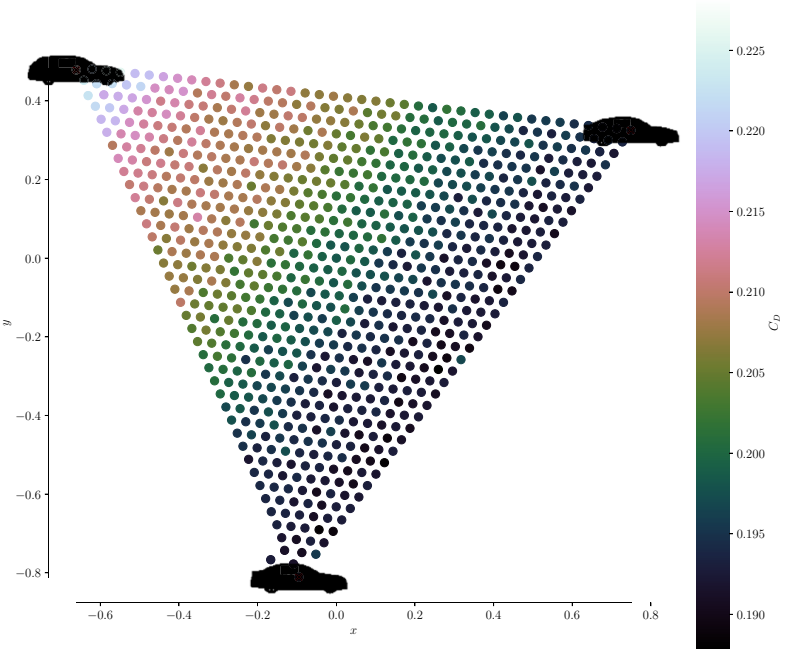


Figure 12. *Multidimensional scaling (MDS) of the DrivAerCH3 dataset. Each sample is colored by drag coefficient obtained from LES. Silhouettes indicate the basis samples.*

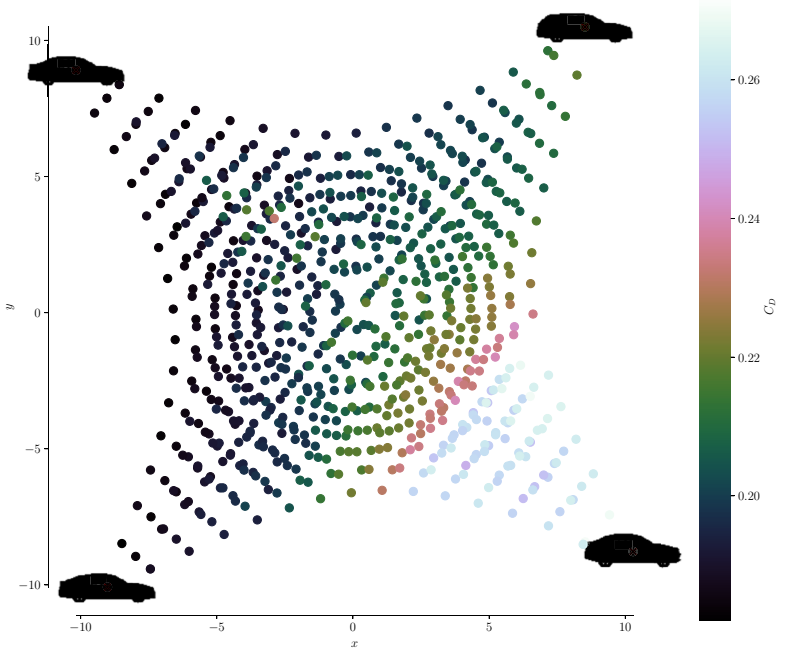


Figure 13. *DrivAerCH4 dataset. See Figure 12 for details.*

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

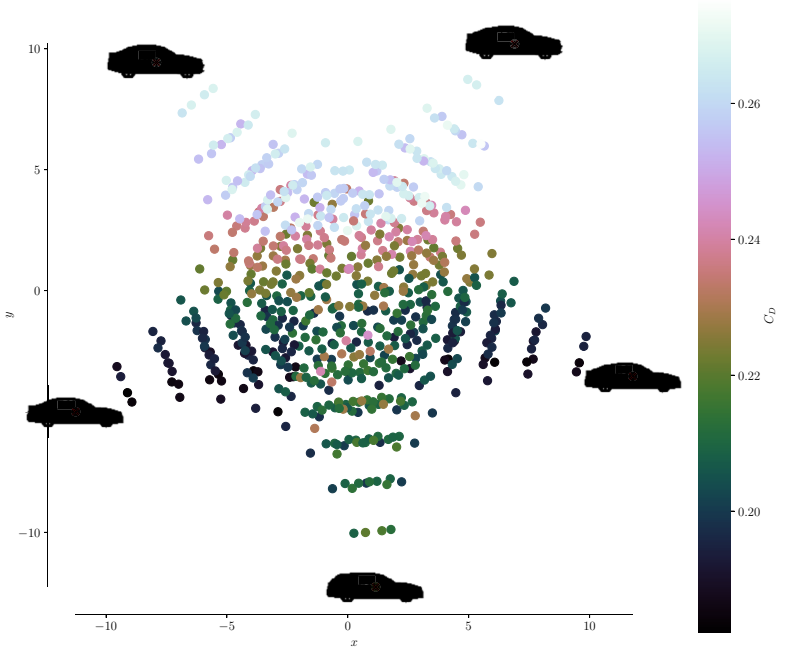


Figure 14. *DrivAerCH5 dataset. See Figure 12 for details.*

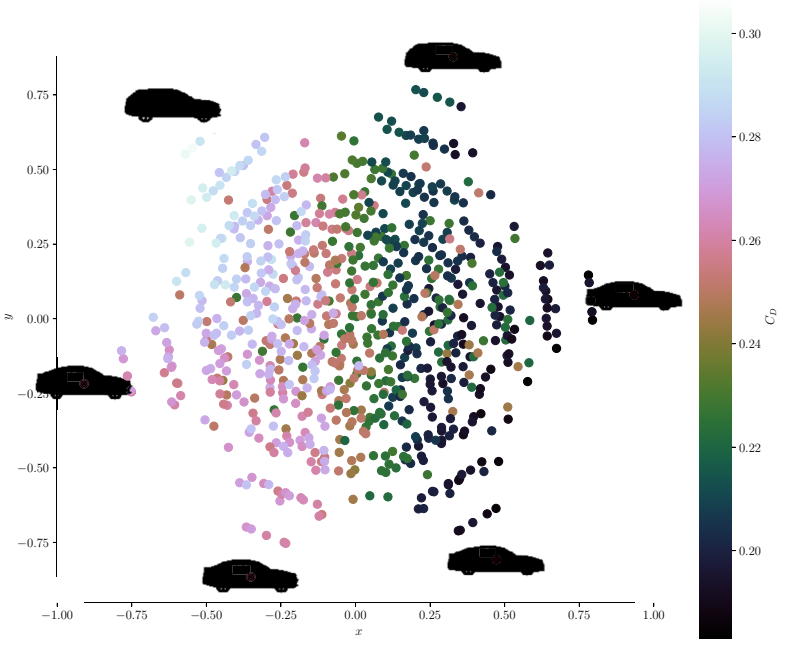


Figure 15. *DrivAerCH6 dataset. See Figure 12 for details.*

The U-Nets are also used to make surface pressure predictions on the four datasets. Figure 17 shows the prediction of one sample from the test set from each of the models. A similar inverse correlation with diversity and accuracy is seen. The model trained on the DrivAerCH3 dataset reconstructs the fine details

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

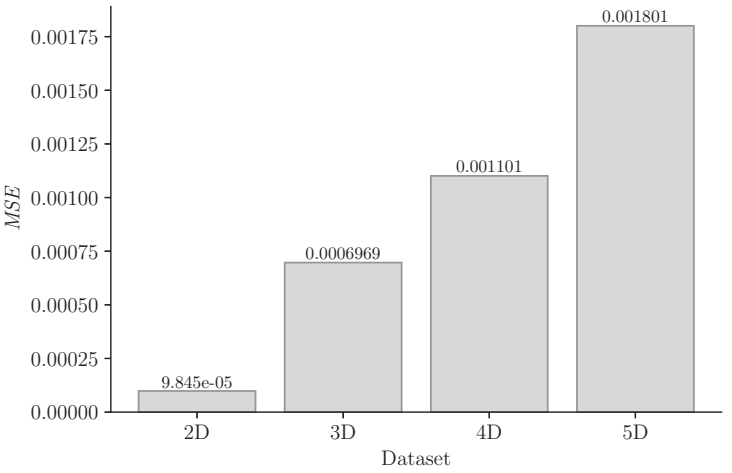


Figure 16. *Mean absolute error in drag counts for CNNs trained on different datasets when predicting the aerodynamic drag. Left to right: DrivAerCH3, DrivAerCH4, DrivAerCH5, and DrivAerCH6.*

of the low-pressure zones created by the vortices released from the roofline–rear window junction (Heft et al., 2012b); this accurate representation gradually degrades until there is almost no fine-grained structure left in the predictions of the model trained on the most diverse DrivAerCH6 dataset.

4.2. A priori evaluation of DrivAerCH3

Beyond aggregate accuracy, the simplex structure of the datasets allows for a spatially resolved analysis of where predictions succeed or fail. One of the benefits of generating data within a simplex is that it allows for direct visualization of different samples in relation to each other. It also allows for the study of the accuracy in interpolation and extrapolation tasks in a controlled manner. Typically, extrapolation testing involves making inference on samples from out of the training set distribution, with the idea being that good predictive performance on these unseen samples suggests that the network is generalizing well. Using the ShapeCH procedure, we can make the process systematic, and introduce an *a priori* measure of extrapolation distance based on the inputs alone.

To do this, the following training–testing split of the DrivAerCH3 database is considered: the points inside the incircle of the barycentric triangle are used for training, and those outside are used for testing (see Figure 18). Using this split, the testing set and training set are not drawn from the same distribution; in fact, the two sets are completely disjoint on the map. However, it is vital to also note that the testing set is not completely disconnected from the training set: the two differ only by the ranges of the two free parameters used to generate the convex hull from the same basis cases. With this in mind, it becomes apparent that those samples in the testing set that lie closer to the circumference of the incircle are "closer" to the training set in the feature space, and are expected to be less challenging to predict, with the amount of extrapolation increasing with increasing distance away from the incircle. Thus, it is possible to assign an "extrapolation distance" to each sample in the testing set, simply computed as the Euclidean distance from the circumference. This is shown in Figure 19, expressed as a percentage ranging from 0% on the circle to 100% at the three vertices.

Next, a CNN to predict C_D (with similar specifications to the one before) is trained on the data inside the incircle. The expectation is that the network prediction error on the testing set will mirror the contours of Figure 19. The error is shown in Figure 20. It is seen that the lower left vertex, corresponding to the estateback basis case, has the spatial error distribution that is expected, with the error increasing radially outward from the incircle and reaching a maximum at the vertex. However, this behavior is not seen in the vicinity of the other two vertices. The key to resolving this anomalous behavior is to recognize that the

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

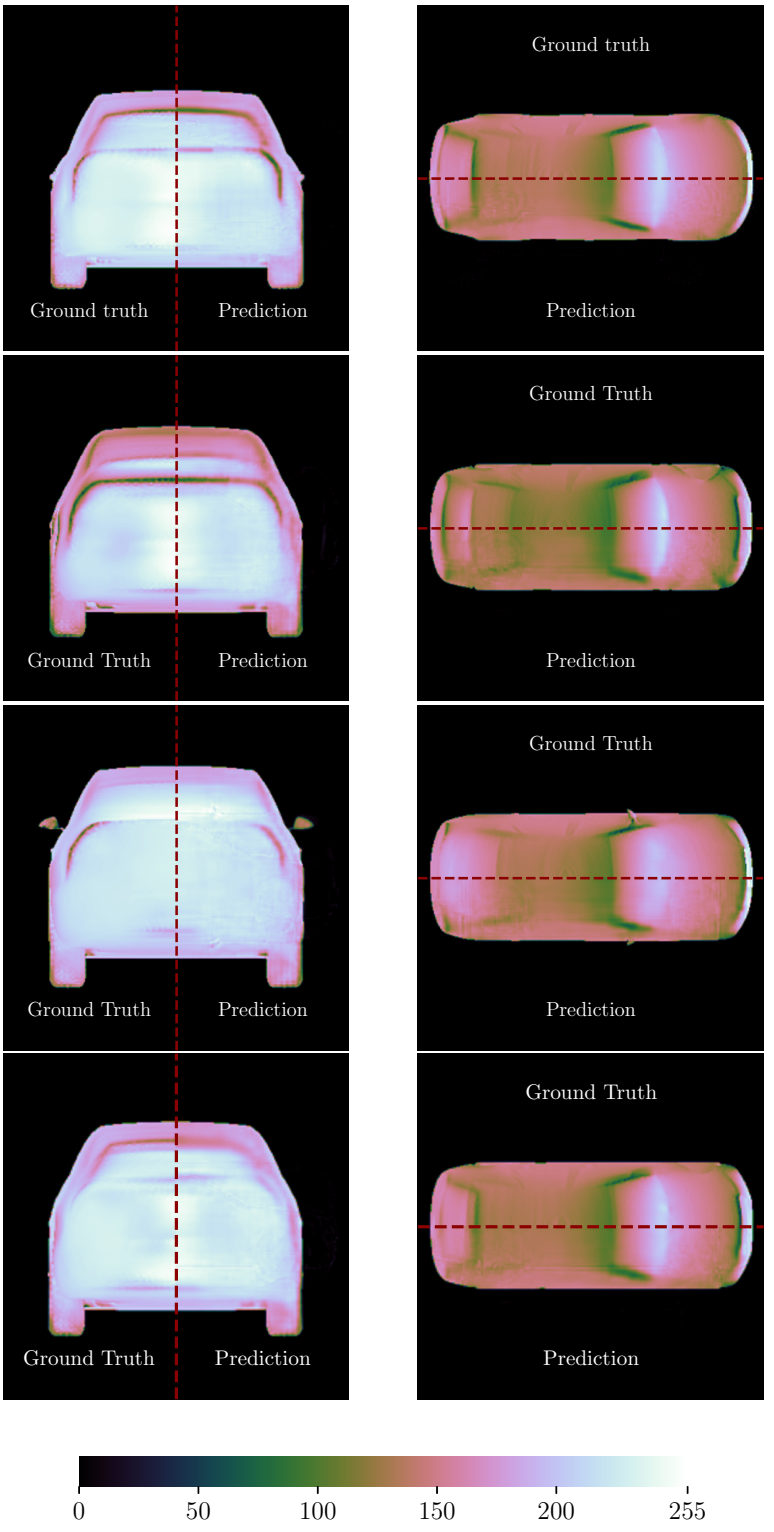


Figure 17. *Surface pressure coefficient c_p predictions for models trained on DrivAerCH3 (first row), DrivAerCH4 (second row), DrivAerCH5 (third row), and DrivAerCH6 (fourth row), respectively.*

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

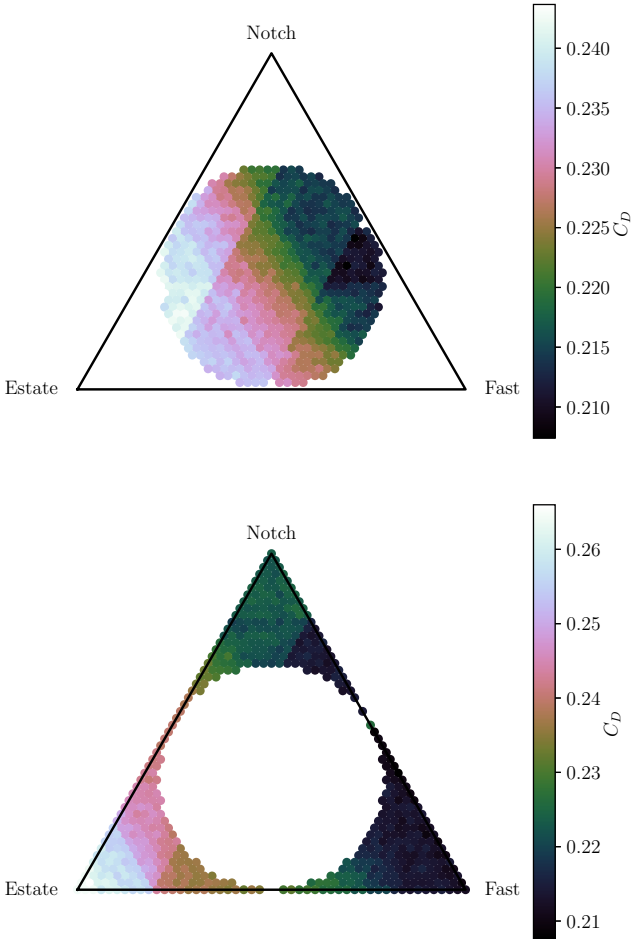


Figure 18. *Extrapolation study: training set (top) and test set (bottom), shaded by C_D.*

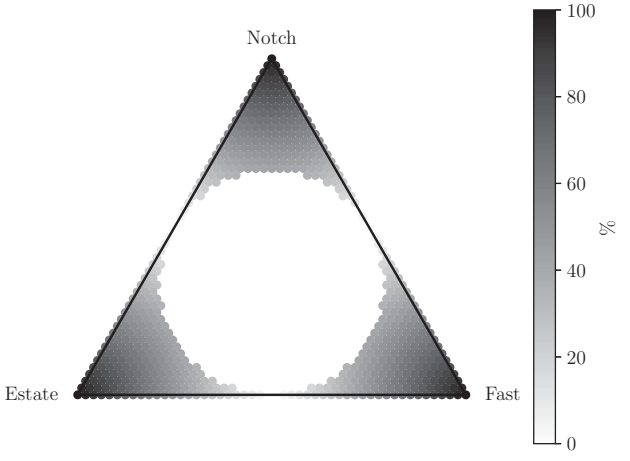


Figure 19. *Extrapolation distance (%).*

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

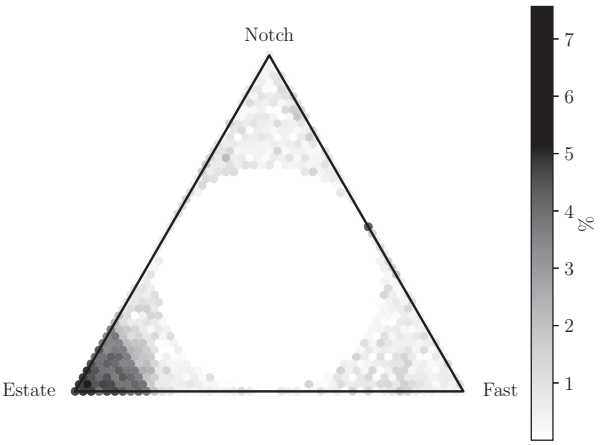


Figure 20. *Test set relative error from extrapolation study.*

extrapolation distance does not account for the actual basis cases assigned to each vertex of the simplex. In other words, Figure 20 is independent of the database. It is clear that a full picture of extrapolation distance must include the effect of varying basis cases, so that if a certain vertex (or basis case) is much further away than the others in feature space, this is reflected by a corresponding increase in the extrapolation distance toward that vertex. This may be alternatively thought of as a rescaling of the sides of the triangle to reflect distances between the basis configurations. To do this, we begin by computing the pairwise distance matrix (see Equation (1)) for the DrivAerCH3 database:

$$\begin{array}{c} \\ E \\ F \\ N \end{array} \begin{array}{c} \begin{array}{ccc} E & F & N \end{array} \\ \begin{pmatrix} 0 & 0.875 & 1.12 \\ 0.875 & 0 & 0.304 \\ 1.12 & 0.304 & 0 \end{pmatrix} \end{array} = D_{pq} \times 10^4. \tag{3}$$

The matrix has a zero leading diagonal by construction, and the off-diagonal elements represent pairwise distances between basis cases. Here, the letter E denotes the estateback, F the fastback, and N the notchback. It is observed that the distance between the notchback and the fastback (0.304×10^{-4}) is approximately a factor of three smaller than the distance between the estateback and the fastback, and a factor of four smaller than the distance between the notchback and the estateback. This information provides the coefficients required to rescale the extrapolation distances from Figure 19, in order to account for the relative differences between the basis cases in feature space. Upon rescaling the distances with the values from the earlier matrix, the modified distance map shown in Figure 21 is obtained. The map now paints a much more accurate picture of the extent to which the network is extrapolating, and it is clear that the higher accuracy of the predictions in the vicinity of the fastback and notchback vertices is due to the low feature space variation in those geometries, relative to the estateback vertex.

4.3. A priori evaluation of DrivAerCH5

The earlier exercise is repeated for the higher diversity DrivAerCH5 dataset, which has two SUV configurations in addition to the three sedans (see Table 6). The MDS projection in 2D is again used to divide the 715 samples into 357 training and 358 test set samples (see Figure 21). This is unlike the division of the barycentric triangle for the previous section in that it is sensitive to the numerical computation of the MDS (Figure 22). The training is done using the CNN architecture from Section 3.5. The results of the training are shown in Figure 23. The error plot shows a concentration of poor predictions

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

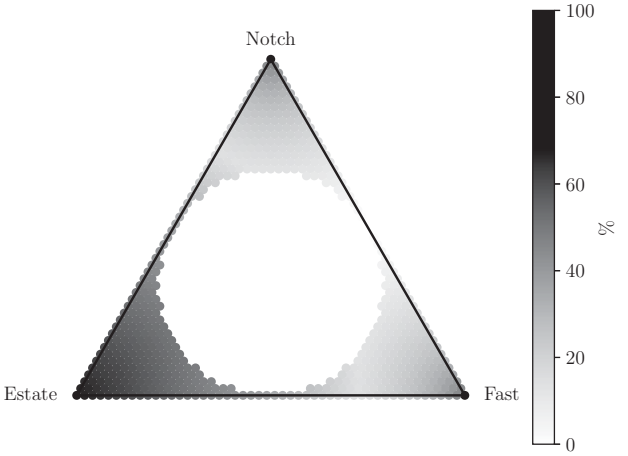


Figure 21. *Test set relative error from extrapolation study rescaled by basis case diversity.*

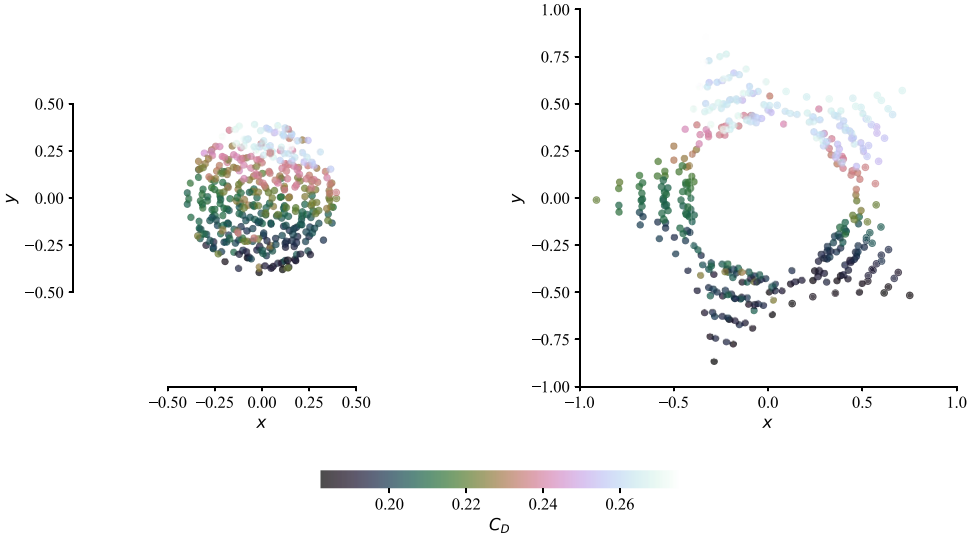


Figure 22. *Controlled extrapolation test of C_D prediction on the DrivAerCH5 dataset; left: MDS projection of training set, colored by C_D value; right: testing set.*

in the subsimplex with the vertices corresponding to the SUV models. As before, we compute the rescaled extrapolation distances using the SDFs of the five basis geometries, and plot them on the MDS projection. It is seen that the regions of high extrapolation distance correlate well with the inference error plot again. One notes that the prediction error is in fact lower than that expected from the extrapolation distances. The reason is because the diversity metric is conditioned on the input (geometric) space to allow for a priori assessments of the test samples. In this case, the results indicate a more expressive geometric space than the corresponding output drag space, so that the model performs better than expected in regions with predominantly the sedan configurations.

The preceding analysis shows that the simplex geometry, combined with the diversity metric, enables *a priori* identification of regions where prediction is difficult. A natural follow-up is to ask whether a *global* relationship between dataset characteristics and prediction error exists.

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

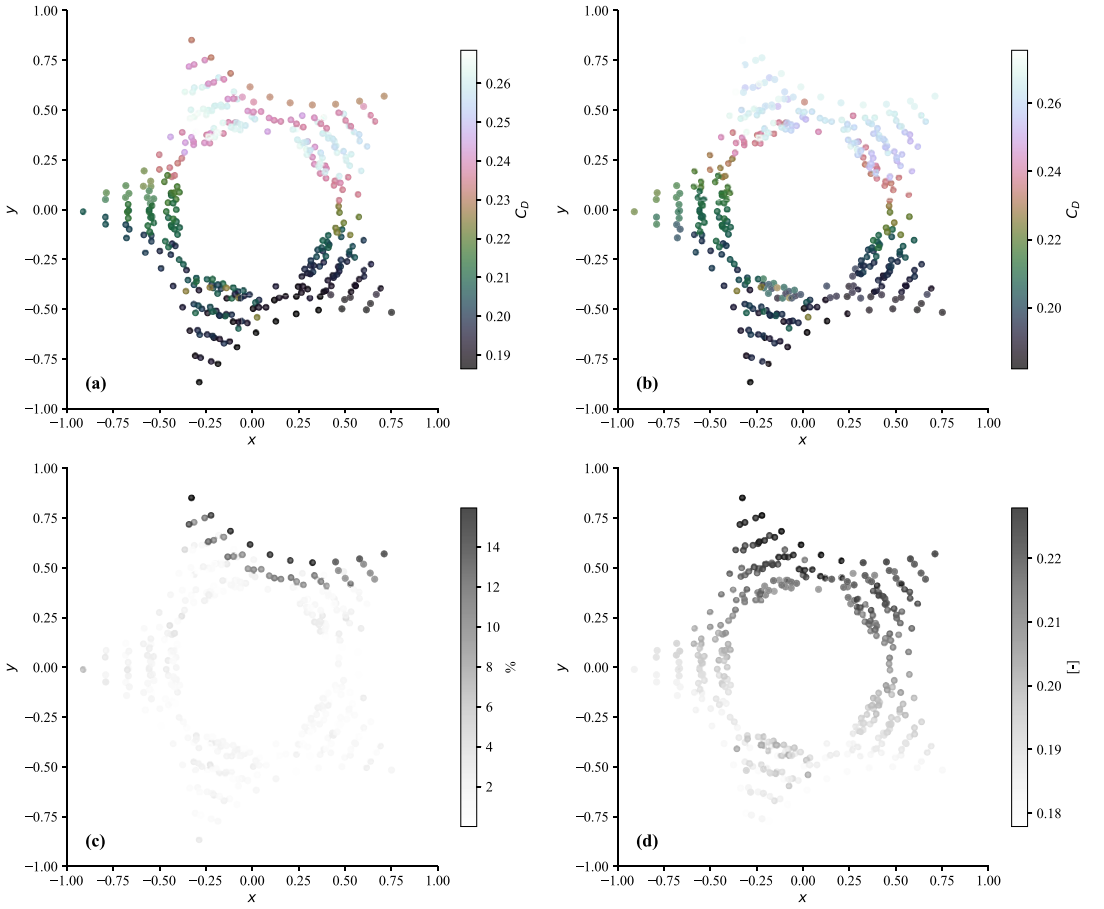


Figure 23. *Controlled extrapolation test of C_D prediction on the DrivAerCH5 dataset. MDS projections of: (a) C_D network predictions, (b) C_D ground truth values, (c) relative error, (d) extrapolation distances computed from diversity metric.*

4.4. Scaling experiments

The qualitative observation that accuracy degrades with diversity motivates a quantitative investigation. In this section, a series of controlled experiments is designed to isolate the effects of dataset size m and diversity M on the prediction error. The goal is to determine whether a scaling law exists that collapses the error from all experiments onto a single curve parameterized by M and m alone.

Recall from Table 6 that M, g, m, and the simplex dimension p are not independent: for a uniformly sampled p-simplex with g points per edge, $m = \binom{g+p}{p}$. Furthermore, the normalization of the diversity metric (Equation (2)) couples M to p through the number of basis-case pairs. Consequently, it is not possible to vary each of M, g, m, and p independently. Rather than attempting to disentangle these coupled quantities, the approach adopted here is empirical: six families of experiments are designed, each targeting a different combination of the design variables, and a scaling law is sought that collapses all of them simultaneously.

The experiment families are designed to isolate these effects by holding one quantity constant at a time. *Constant-size* experiments fix m and vary the simplex (and thus M), revealing the effect of diversity alone. *Constant-diversity* experiments fix the simplex (and thus M) and extract barycentric subgrids at different grid densities g, varying size through the combinatorial constraint. *Constant-density* experiments fix the

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

grid density ($g = 3$ and $g = 5$) across DrivAerCH4–CH6, so that m and M both grow with the simplex dimension. All three types are summarized in Table 7.

All 23 experiments use the CNN architecture described in Section 2.4. To ensure that the scaling behavior is not an artifact of a single network configuration, each experiment is repeated with eight hyperparameter settings drawn from a $2 \times 2 \times 2$ grid: number of convolutional layers $\in \{3,4\}$, channel width $\in \{64,128\}$, and batch size $\in \{16,32\}$, yielding networks with approximately 218,000 to 690,000 trainable parameters. The learning rate (10^{-4}) and number of epochs (500) are held fixed. For each of the $23 \times 8 = 184$ configurations, fivefold cross-validation is performed, so that a total of 920 networks are trained from scratch. The reported MAE for each experiment is the fivefold mean of the best-performing hyperparameter configuration, and the $\pm$ bounds in Table 7 are 95% confidence intervals computed from the eight configuration means using a t distribution with seven degrees of freedom. These intervals isolate the sensitivity of the prediction error to the network architecture. In all cases, the MAE is evaluated on the held-out test fold, so that every sample is predicted exactly once by a network that never saw it during training.

Table 7. *Summary of scaling experiments*

Experiment	p	g	m	M	MAE
Constant size ($m = 715$)					
CH3	2	39	715	5.253×10^{-8}	11.8 ± 0.7
CH4	3	15	715	1.047×10^{-7}	16.6 ± 0.4
CH5	4	9	715	1.067×10^{-7}	16.8 ± 0.2
CH6	5	7	715	1.012×10^{-7}	14.7 ± 0.5
Constant size ($m = 300$)					
CH3	2	39	300	5.253×10^{-8}	11.2 ± 0.3
CH4	3	15	300	1.047×10^{-7}	16.9 ± 0.4
CH5	4	9	300	1.067×10^{-7}	16.8 ± 0.5
CH6	5	7	300	1.012×10^{-7}	16.1 ± 0.7
Constant size ($m = 500$)					
CH3	2	39	500	5.253×10^{-8}	11.8 ± 0.4
CH4	3	15	500	1.047×10^{-7}	15.7 ± 0.9
CH5	4	9	500	1.067×10^{-7}	16.3 ± 0.6
CH6	5	7	500	1.012×10^{-7}	16.4 ± 0.5
Const. diversity (*CH3*)					
$g = 6$	2	6	21	5.253×10^{-8}	20.8 ± 9.4
$g = 12$	2	12	78	5.253×10^{-8}	13.1 ± 0.7
$g = 24$	2	24	300	5.253×10^{-8}	11.4 ± 0.5
$g = 39$	2	39	820	5.253×10^{-8}	12.0 ± 0.7
$g = 50$	2	50	1,270	5.253×10^{-8}	9.8 ± 0.3
Const. density ($g = 3$)					
CH4	3	3	20	1.047×10^{-7}	21.4 ± 2.8
CH5	4	3	35	1.067×10^{-7}	21.3 ± 2.1
CH6	5	3	56	1.012×10^{-7}	26.3 ± 2.1
Const. density ($g = 5$)					
CH4	3	5	56	1.047×10^{-7}	26.6 ± 0.7
CH5	4	5	126	1.067×10^{-7}	19.7 ± 1.7
CH6	5	5	252	1.012×10^{-7}	17.7 ± 0.7

Note. Here p is the simplex dimension, g the grid density (points per edge), m the number of training samples, M the diversity metric, and MAE the mean absolute error in drag counts (one drag count $= \Delta C_D = 0.001$) for the best-performing hyperparameter configuration. The $\pm$ range denotes the 95% confidence interval over eight hyperparameter configurations (see text), quantifying network sensitivity.

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

4.5. Scaling law

To quantify the impact of dataset size, density, and diversity, a relationship between the prediction error and the dataset characteristics is sought. The most general form, involving all four characterization variables, is

$$\varepsilon = \varepsilon_0 M^a m^b g^c p^d, \tag{4}$$

where ε denotes the MAE, ε_0 absorbs the irreducible loss and order-unity coefficients, and a, b, c, d are exponents to be determined. Taking logarithms converts (4) to a linear regression, and ordinary least squares is applied to the 23 data points from Table 7.

The full four-variable fit yields $R^2 = 0.88$, but the exponents on g and p are not statistically significant ($p_g = 0.35$, $p_p = 0.54$); their information is already captured by M and m through the combinatorial constraint $m = \binom{g+p}{p}$, which makes m, g, and p collinear. Conversely, fitting m, g, and p without M yields $R^2 = 0.82$ with no individually significant exponent ($p_m = 0.80, p_g = 0.09, p_p = 0.84$), confirming that M carries information that the other variables cannot replace. Including only M and m gives

$$a = 1/2,\ b = -1/6,\ R^2 = 0.87, \tag{5}$$

with negligible loss in explanatory power and both exponents highly significant ($p < 10^{-3}$). The grid density g and simplex dimension p are therefore dropped, yielding the scaling law:

$$\boxed{\varepsilon = \varepsilon_0 M^{1/2} m^{-1/6}}. \tag{6}$$

The quality of the scaling law is shown in Figure 24. A composite scaling variable $\xi \propto M^{1/2} m^{-1/6}$ is formed, and the MAE of all 23 experiments is plotted against ξ. Error bars indicate the standard error of the

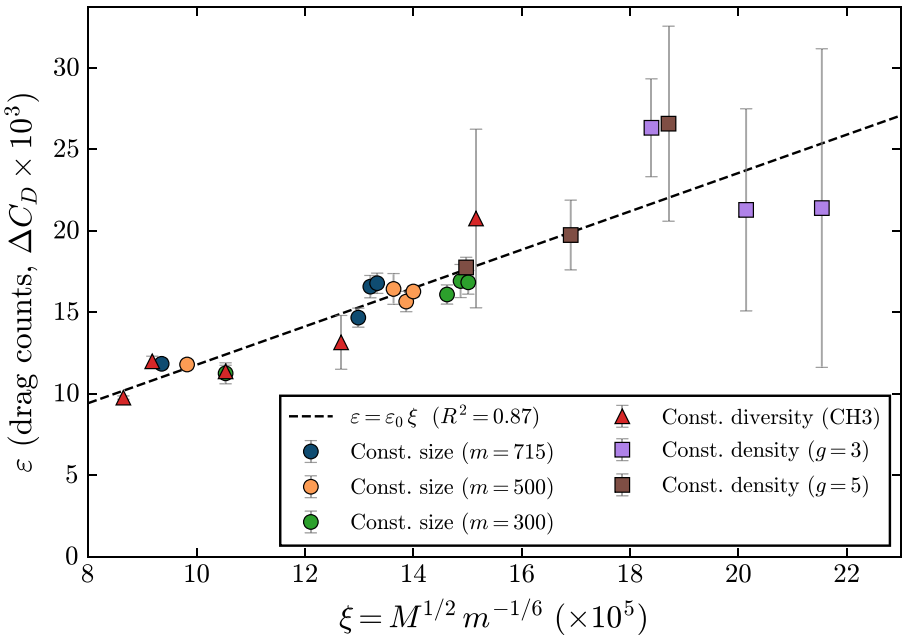


Figure 24. *All 23 experiments from Table 7 plotted against the composite variable $\xi \propto M^{1/2} m^{-1/6}$ (scaled by 10^5 for readability). Colors distinguish the six experiment families; markers denote the constraint type (circles: constant size; squares: constant density; triangles: constant diversity). Error bars show the standard error of the mean MAE across the five cross-validation folds of the best-performing network. The dashed line is the fit from Equation (6).*

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

mean across the five cross-validation folds, which is largest for the low-m experiments (up to ± 10 drag counts for $m = 20$) and negligible for well-sampled datasets. The narrow confidence intervals reported in Table 7 confirm that the collapse is not an artifact of a specific network architecture. The data from all experiment families (spanning different simplex dimensions, densities, sizes, and diversities) fall on a single line, confirming that the prediction error is governed by two quantities alone: the diversity of the basis shapes and the number of training samples.

The dependence on dataset size ($m^{-1/6}$) is considerably weaker than the dependence on diversity ($M^{1/2}$): a tenfold increase in training samples reduces the MAE by roughly 33%, whereas the same factor in diversity increases it by more than threefold. This asymmetry suggests that, beyond a certain dataset size, further gains are better achieved by controlling the diversity of the basis shapes than by adding more samples.

The exponent $b = -1/6$ on dataset size is notably weaker than the $m^{-1/2}$ convergence rate of Monte Carlo estimators based on independent samples. This is expected: the training geometries are not independent but are generated by barycentric interpolation of a small number of basis shapes, introducing strong correlations among samples. This sub-MC convergence is a manifestation of oversampling in synthetic datasets and suggests that the scaling law (6) may not generalize to parametrizations with weaker intersample correlations, where a steeper exponent could be expected. Neither the grid density g nor the simplex dimension p appear explicitly in (6). This is not because they are irrelevant, but because their effects are fully absorbed by M and m. The diversity metric M already encodes p through its normalization over $p(p+1)$ basis-case pairs, and the sample count m absorbs both g and p through the combinatorial constraint $m = \binom{g+p}{p}$. As shown earlier, including g and p as additional regressors, improves R^2 by less than 0.01, confirming that these variables carry no independent information beyond what is already captured by M and m.

Unlike neural scaling laws for large models (Kaplan et al., 2020), which include the number of model parameters N as an explicit variable, Equation (6) omits N. Each of the 23 experiments in Table 7 is repeated with eight network configurations spanning a threefold range in parameter count (218,000–690,000), and the resulting variation in MAE is small compared with the interexperiment differences: the median 95% confidence interval from the eight configurations is 0.7 drag counts, while the dataset-driven MAE ranges from 9.8 to 26.6 drag counts. At the relatively modest network and dataset sizes considered here, network capacity is therefore a secondary effect, and the prediction error is dominated by the dataset characteristics M and m.

The implication of the earlier for practitioners is clear: when designing a dataset for solving shape-to-performance studies, the two decisions that matter are (i) how geometrically diverse the basis shapes should be and (ii) how many samples to generate. The choice of simplex dimension or grid density is secondary, as long as the resulting m is sufficient. An explicit measure of shape diversity (not necessarily what is suggested here) is the most critical tool to evaluate existing datasets or generating new ones.

5. Conclusions

In this study, a conceptual framework for characterizing datasets used for building machine learning models is introduced. The application revolves around a shape-to-aerodynamic prediction task relevant to automotive design, though it is general in nature. Specifically, we consider size, density, and diversity as key metrics for characterizing the dataset. Size and density refer to the total number of samples and the coverage of the shape parametrization, respectively. Diversity is, instead, a quantitative metric describing the overall variety in the dataset.

The diversity measure relies on the signed distance function and the choice of few basis geometries. Only the input geometrical shapes are considered to demonstrate that the diversity in the input space is correlated with the diversity in the output space (Figure 11). This enables to estimate the "learning complexity" a priori in the absence of ground truth data (i.e., before extensive resources are devoted to generate simulation data).

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

Four different datasets are generated using a different number of basis configurations (from 3 to 6); the datasets are designed to have fixed size but varying density and diversity. When data-driven models are trained on these datasets, it is observed that there is an inverse correlation between model accuracy and dataset diversity. This observation is made precise by an empirical scaling law, $\varepsilon = \varepsilon_0 M^{1/2} m^{-1/6}$, which collapses 23 controlled experiments spanning different simplex dimensions, grid densities, and dataset sizes onto a single curve ($R^2 = 0.87$). The scaling reveals that diversity has a substantially stronger effect on prediction error than dataset size, offering a practical guideline for dataset design. The proposed framework and characterization allows for meaningful comparisons between different databases.

Finally, geometries generated using the proposed approach lend themselves to a natural low-dimensional encoding given by the convex interpolation weights (or, equivalently, coordinates within the simplex). This allows for the definition of an extrapolation distance that quantifies how close new samples are to the existing training set. This is demonstrated in a controlled manner by a novel training–testing split, and examining the correlation between extrapolation distance and model inference accuracy. It is seen that the correlation is uncovered when the diversity of the basis cases is properly accounted for.

Taken together, these results underscore that *data for learning* must precede *learning from data*: before investing in model architecture or training, one must ensure that the dataset itself is adequate. The characterization framework introduced here provides the tools to make that assessment rigorously.

Data availability statement. The DrivAerCH3 dataset is available at the following repository: https://huggingface.co/datasets/mulligatawnysoup/DrivAerCH3 (Benjamin and Iaccarino, 2025a). The DrivAerCH4, DrivAerCH5, and DrivAerCH6 datasets are available at https://huggingface.co/datasets/mulligatawnysoup/DrivAerCHX (Benjamin and Iaccarino, 2025b). The ShapeCH repository can be found at https://code.stanford.edu/markben/ShapeCH. Other scripts used for the experiments in this work may be obtained by communicating with the authors directly.

Acknowledgments. The authors acknowledge useful discussions with Dr Frank Ham and Kun Lu at Cadence Design Systems, San Jose, CA. The computations performed in this study were performed on the NREL Kestrel HPC Cluster in Golden, CO. The authors acknowledge the use of LLM-based coding assistants during the conduct of this work.

Author contribution. Data curation: M.B., G.I.; Formal analysis: M.B.; Investigation: M.B., G.I.; Methodology: M.B., G.I.; Project administration: M.B., G.I.; Resources: M.B., G.I.; Software: M.B.; Supervision: M.B., G.I.; Writing - original draft: M.B.; Conceptualization: G.I.; Funding acquisition: G.I.; Validation: G.I.; Writing - review & editing: G.I.

Funding statement. This work was funded by the Franklin & Caroline Johnson Fellowship at the Stanford University.

Competing interests. The authors are aware of no competing interests that could potentially have impacted the conclusions drawn from this work.

References

Ahmed SR, **Ramm G and Faltin G** (1984) Some salient features of the time-averaged ground vehicle wake. *SAE Transactions 93*, 473–503.

Ambo K, **Nagaoka H**, **Philips DA**, **Ivey C**, **Brès GA and Bose ST** (2020) Aerodynamic force prediction of the laminar to turbulent flow transition around the front bumper of the vehicle using dynamic-slip wall model LES. In *AIAA Scitech 2020 Forum*. Reston, VA: American Institute of Aeronautics and Astronautics, AIAA Paper 2020-0036.

Ashton N and Revell A (2015) Comparison of RANS and DES methods for the drivaer automotive body. In *SAE 2015 World Congress & Exhibition*. Detroit, MI: SAE International, pp. 1–10.

Ashton N, **Mockett C**, **Fuchs M**, **Fliessbach L**, **Hetmann H**, **Knacke T**, **Schonwald N**, **Skaperdas V**, **Fotiadis G**, **Walle A**, **Hupertz B and Maddix D** (2024) Drivaerml: High-fidelity computational fluid dynamics dataset for road-car external aerodynamics. *arxiv.org* preprint, arXiv:2408.11969. Available at: https://arxiv.org/abs/2408.11969.

Aultman M, **Wang Z**, **Auza-Gutierrez R and Duan L** (2022) Evaluation of CFD methodologies for prediction of flows around simplified and complex automotive models. *Computers & Fluids 236*, 105297.

Baque P, **Remelli E**, **Fleuret F and Fua P** (2018) Geodesic convolutional shape optimization. In Dy J and Krause A (eds), *Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research*. PMLR, pp. 472–481.

Benjamin M and Iaccarino G (2025a) DrivAerCH3 [dataset]. *Hugging Face.* Available at https://huggingface.co/datasets/mulligatawnysoup/DrivAerCH3 (accessed 6 July 2026).

Benjamin M and Iaccarino G (2025b) DrivAerCHX: DrivAerCH4, DrivAerCH5, and DrivAerCH6 [dataset]. *Hugging Face.* Available at https://huggingface.co/datasets/mulligatawnysoup/DrivAerCHX (accessed 6 July 2026).

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

Benjamin M and Iaccarino G (2025c) A systematic dataset generation technique applied to data-driven automotive aerodynamics. *APL Machine Learning 3*(1). https://doi.org/10.1063/5.0233367.

Benjamin M, **Saetta E**, **Ivey C**, **Bose S**, **Ham F and Iaccarino G** (2023) *Parametric representation and surrogate modeling of drag for a realistic automotive geometry*. Stanford, CA: Center for Turbulence Research Annual Research Briefs.

Brazdil P, **van Rijn JN**, **Soares C**, **Vanschoren J**, **Brazdil P**, **van Rijn JN**, **Soares C and Vanschoren J** (2022) Dataset Characteristics (Metafeatures). In *Metalearning: Applications to Automated Machine Learning and Data Mining*. Cham: Springer International Publishing, pp. 53–75.

Brès G, **Bose S**, **Emory M**, **Ham F**, **Schmidt O**, **Rigas G and Colonius T** (2018) Large-eddy simulations of co-annular turbulent jet using a Voronoi- based mesh generation framework. In *AIAA Paper 2018-3302*. Reston, VA: American Institute of Aeronautics and Astronautics.

Chang AX, **Funkhouser T**, **Guibas L**, **Hanrahan P**, **Huang Q**, **Li Z**, **Savarese S**, **Savva M**, **Song S**, **Su H**, **Xiao J**, **Yi L and Yu F** (2015) Shapenet: An information-rich 3D model repository. arXiv Preprint, arXiv:1512.03012.

Choquette J (2023) NVIDIA Hopper H100 GPU: Scaling performance. *IEEE Micro 43*(3), 9–17. https://doi.org/10.1109/MM.2023.3256796.

Cox TF and Cox MAA (2000) *Multidimensional Scaling*. Boca Raton, FL: Chapman and Hall/CRC.

Cui Y, **Gu Z**, **Mahajan D**, **van der Maaten L**, **Belongie SJ and Lim S-N** (2019) Measuring Dataset Granularity. *CoRR*, abs/1912.10154.

Elrefaie M, **Dai A and Ahmed F** (2024a) DrivAerNet: A parametric car dataset for data-driven aerodynamic design and graph-based drag prediction. *Journal of Mechanical Design 147*(4), 041712.

Elrefaie M, **Morar F**, **Dai A and Ahmed F** (2024b) 'DrivAerNet++: A large-scale multimodal car dataset with computational fluid dynamics simulations and deep learning benchmarks', in Globerson, A, Mackey, L, Belgrave, D, Fan, A, Paquet, U, Tomczak, J and Zhang, C (eds) *Advances in Neural Information Processing Systems 37*. Red Hook, NY: Curran Associates, Inc., pp. 499–536. doi:10.52202/079017-0016.

Garcia-Fernandez R, **Portal-Porras K**, **Irigaray O**, **Ansa Z and Fernandez-Gamiz U** (2023) CNN-based flow field prediction for bus aerodynamics analysis. *Scientific Reports 13*(1), 21213. https://doi.org/10.1038/s41598-023-48419-4.

Goc KA, **Lehmkuhl O**, **Park GI**, **Bose ST and Moin P** (2021) Large eddy simulation of aircraft at affordable cost: a milestone in computational fluid dynamics. *Flow 1*, E14. https://doi.org/10.1017/flo.2021.17.

Gunpinar E, **Coskun UC**, **Ozsipahi M and Gunpinar S** (2019) A generative design and drag coefficient prediction system for sedan car side silhouettes based on computational fluid dynamics. *Computer-Aided Design 111*, 65–79. https://doi.org/10.1016/j.cad.2019.02.003.

Hanocka R, **Hertz A**, **Fish N**, **Giryes R**, **Fleishman S and Cohen-Or D** (2019) MeshCNN: A network with an edge. *ACM Transactions on Graphics (TOG) 38*, 90:1–90:12.

Heft A, **Indinger T and Adams N** (2012a) Introduction of a new realistic generic car model for aerodynamic investigations. *SAE Tech. Paper* 2012-01-0168. Warrendale, PA: SAE International.

Heft AI, **Indinger T and Adams NA** (2012b) Experimental and numerical investigation of the DrivAer model. In *Fluids Engineering Division Summer Meeting*, Vol. *44755*. New York, NY: American Society of Mechanical Engineers, pp. 41–51.

Jacob, SJ, **Mrosek, M**, **Othmer, C and Köstler, H** (2022) 'Deep learning for real-time aerodynamic evaluations of arbitrary vehicle shapes', *SAE International Journal of Passenger Vehicle Systems*, *15*(2), pp. 77–90. doi:10.4271/15-15-02-0006.

Kaplan J, **McCandlish S**, **Henighan T**, **Brown TB**, **Chess B**, **Child R**, **Gray S**, **Radford A**, **Wu J and Amodei D** (2020) Scaling laws for neural language models. arXiv Preprint, arXiv:2001.08361.

Kipf TN and Welling M (2017) Semi-supervised classification with graph convolutional networks. arXiv Preprint, arXiv:1609.02907.

Kolmogorov AN (1941) The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers. *Doklady Akademii Nauk SSSRDokl. Akad. SSSR 30*, 299–303.

Le Good GM and Garry KP (2004) On the use of reference models in automotive aerodynamics. In *SAE World Congress*. Warrendale PA: SAE International, SAE Technical Paper 2004-01-1308. doi:10.4271/2004-01-1308.

LeCun Y, **Bengio Y and Hinton G** (2015) Deep learning. *Nature 521*(7553), 436–444. https://doi.org/10.1038/nature14539.

Li Z, **Kovachki NB**, **Azizzadenesheli K**, **Liu B**, **Bhattacharya K**, **Stuart AM and Anandkumar A** (2020) Fourier neural operator for parametric partial differential equations. *CoRR*, abs/2010.08895.

Li Z, **Kovachki NB**, **Choy C**, **Li B**, **Kossaifi J**, **Otta SP**, **Nabian MA**, **Stadler M**, **Hundt C**, **Azizzadenesheli K and Anandkumar A** (2023) Geometry-informed neural operator for large-scale 3D PDEs. In Oh A, Naumann T, Globerson A, Saenko K, Hardt M and Levine S (eds) *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023) 36*. Red Hook, NY: Curran Associates, Inc., pp. 35836–35854.

Lin T-Y, **Dollár P**, **Girshick R**, **He K**, **Hariharan B and Belongie S** (2017) Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. Honolulu, HI: IEEE, pp. 2117–2125.

Liu Z, **Tang H**, **Lin Y and Han S** (2019) Point-voxel CNN for efficient 3D deep learning. In *Advances in Neural Information Processing Systems*. Red Hook, NY: Curran Associates, Inc., pp. 963–973.

Mammola S, **Carmona CP**, **Guillerme T and Cardoso P** (2021) Concepts and applications in functional diversity. *Functional Ecology 35*(9), 1869–1885. https://doi.org/10.1111/1365-2435.13882.

Maturana D and Scherer S (2015) VoxNet: A 3D convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Piscataway, NJ: IEEE, pp. 922–928.

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press

Nagaoka H, **Yenerdag B**, **Ambo K**, **Philips D**, **Ivey C**, **Brès G and Bose S** (2024) Prediction of aerodynamic drag in SUVs with different specifications by using large-eddy simulations. In *InWCX SAE World Congress Experience*. Warrendale, PA: SAE International, SAE Technical Paper 2024-01-2525. doi:10.4271/2024-01-2525.

Qi CR, **Su H**, **Mo K and Guibas LJ** (2017a) PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. Honolulu, HI: IEEE Computer Society, pp. 652–660.

Qi CR, **Yi L**, **Su H and Guibas LJ** (2017b) PointNet++: Deep hierarchical feature learning on point sets in a metric space. In Guyon I, von Luxburg U, Bengio S, Wallach H, Fergus R, Vishwanathan SVN and Garnett R (eds) *Advances in Neural Information Processing Systems (NeurIPS)*. Red Hook, NY: Curran Associates, Inc., pp. 5099–5108.

Qiu J, **Kuang L**, **Wang G**, **Xu Y**, **Cui L**, **Fu S**, **Zhu Y and Zhang R** (2025) Drivaerstar: An industrial-grade CFD Dataset for vehicle aerodynamic optimization. *CoRR*, abs/2510.16857.

Remelli E, **Lukoianov A**, **Richter S**, **Guillard B**, **Bagautdinov T**, **Baque P and Fua P** (2020) Meshsdf: Differentiable iso-surface extraction. In Larochelle H, Ranzato M, Hadsell R, Balcan MF and Lin H (eds), *Advances in Neural Information Processing Systems*, Vol. *33*. Red Hook, NY: Curran Associates, Inc, pp. 22468–22478.

Rios T, **Sendhoff B**, **Menzel S**, **Bäck T and van Stein B** (2019) On the efficiency of a point cloud autoencoder as a geometric representation for shape optimization. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. Piscataway, NJ: IEEE, pp. 791–798.

Rios T, **van Stein B**, **Wollstadt P**, **Bäck T**, **Sendhoff B and Menzel S** (2021) Exploiting local geometric features in vehicle design optimization with 3D point cloud autoencoders. In *2021 IEEE Congress on Evolutionary Computation (CEC)*. Piscataway, NJ: IEEE, pp. 514–521.

Rodi W (1997) Comparison of LES and RANS calculations of the flow around bluff bodies. *Journal of Wind Engineering and Industrial Aerodynamics 69*–71, 55–75. https://doi.org/10.1016/S0167-6105(97)00147-5.

Shur ML, **Spalart PR**, **Strelets MK and Travin AK** (2008) Hybrid RANS-LES approaches for detached eddy simulation of turbulent flows. *Flow, Turbulence and Combustion 81*(3), 451–473.

Song B, **Yuan C**, **Permenter F**, **Arechiga N and Ahmed F** (2023) Weakly supervised AI for efficient analysis of 3D pathology samples. arXiv Preprint, arXiv:2306.06110.

Te G, **Hu W**, **Guo Z and Zheng A** (2018) RGCNN: Regularized graph CNN for point cloud segmentation. arXiv Preprint, arXiv:1806.02952.

Trinh TL, **Chen F**, **Nanri T and Akasaka K** (2024) 3D super-resolution model for vehicle flow field enrichment. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. Piscataway, NJ:IEEE, pp. 5826–5835.

Umetani N and Bickel B (2018) Learning three-dimensional flow for interactive aerodynamic design. *ACM Transactions on Graphics 37*(4), 1–10. https://doi.org/10.1145/3197517.3201325.

Usama M, **Arif A**, **Haris F**, **Khan S**, **Afaq K and Rashid S** (2021) A Data-Driven Interactive System for Aerodynamic and User-Centred Generative Vehicle Design. In *Proceedings of the International Conference on Artificial Intelligence (ICAI 2021)*. Piscataway, NJ: IEEE, pp. 119–127.

Vargiemezis T, **Kanatsoulis C and Gorlé C** (2025) WindMil: Equivariant graph learning for wind loading prediction. arXiv Preprint, arXiv:2511.01226.

Viquerat J and Hachem E (2020) A supervised neural network for drag prediction of arbitrary 2D shapes in laminar flows at low Reynolds number. *Computers and Fluids 210*, 104645. https://doi.org/10.1016/j.compfluid.2020.104645.

Wilcox DC (2006) Turbulence modeling for CFD: RANS and beyond. *Turbulence Modeling for CFD 3*, 1–250.

Wu Z, **Pan S**, **Chen F**, **Long G**, **Zhang C and Yu PS** (2021) A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems 32*(1), 4–24. https://doi.org/10.1109/TNNLS.2020.2978386.

Zhang C, **Tanneberger M**, **Kuthada T**, **Wittmeier F**, **Wiedemann J and Nies JM** (2019) *Introduction of the AeroSUV – A New Generic SUV Model for Aerodynamic Research*. SAE Technical Paper 2019-01-0646. Warrendale, PA: SAE International. doi:10.4271/2019-01-0646.

Cite this article: Benjamin M and Iaccarino G (2026). Learning from data versus data for learning: a shape-to-aerodynamics study. *Data-Centric Engineering*, 7, e29. doi:10.1017/dce.2026.10061

https://doi.org/10.1017/dce.2026.10061 Published online by Cambridge University Press