Parameteric representation and surrogate modeling of drag for a realistic automotive geometry

By M. Benjamin, E. Saetta†, C. Ivey‡, S. Bose‡¶, F. Ham‡ AND G. Iaccarino

1. Motivation

The computation of aerodynamic drag is an expensive component of automotive design, requiring one simulation per design point. It is also an essential component of the design process; since drag force is quadratic in vehicle velocity, it becomes the dominant mode of energy loss at high speed, affecting vehicle top speed, fuel efficiency and stability. 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, typically with a closure model to account for all or most of the turbulent scales that are dynamically important at realistic Reynolds numbers, in a computational domain sufficiently large enough to capture the wake behind the car. Surrogate models are therefore of interest; in particular, given the nature of the governing equations and the complexity of the geometries of interest, we seek data-driven models for drag prediction. The surrogate model learns a mapping between the geometry of the automobile and the drag coefficient. Since drag depends only on the wetted surface of the automobile and not the details of the interior, planar projections of the exterior surface are a natural representation choice for the geometry. Song *et al.* (2023) developed a surrogate model that used isometric views of automobile designs from the ShapeNet database (Chang *et al.* 2015). Projections can be naturally assembled into structured image data that can be processed by convolutional neural networks (CNNs). Unlike unstructured representations such as graphs or point clouds, projections naturally incorporate geometry modifications. In addition, CNNs are easier to train than graph neural networks, the architecture of choice for unstructured representations.

The generation of new designs can likewise be performed using a data–driven surrogate, especially in automotive design where one begins with a baseline geometry and the design space for aerodynamic optimization is constrained by other considerations such as safety. In this context, a surrogate model would seek to learn a compressed representation of the geometry and provide the ability to generate new geometric candidates by interpolation in the compressed space. In this work, we use convolutional autoencoders (CAEs) to compress a dataset of automotive geometries that have parameteric design variations. By interpolating in the latent space of the autoencoder (AE), new designs can be generated, the aerodynamic performance of which can be evaluated by the aforementioned CNN model. The CAE and CNN models in conjunction provide an efficient way of sampling and evaluating designs in the parameter space.

† University of Naples Federico II, Italy
‡ Cadence Design Systems
¶ Institute for Computational and Mathematical Engineering, Stanford

Parameter	Minimum value (m)	Maximum value (m)
Ride (p_1)	0.001	0.025
Spoiler (p_2)	0.1	0.5
Wing mirrors (p_3)	0.1	0.5
Radiator (p_4)	0.02	0.1
Door handles (p_5)	0.02	0.1

TABLE 1. Geometric design parameters and ranges. Ranges are reported as differences from the baseline DrivAer geometry.

2. Methodology

2.1. *Geometry and dataset*

To best represent the complexities of a realistic automobile, we consider the DrivAer model (Heft *et al.* 2012), an open-source midsize passenger car geometry, developed to assess the quality of and contrast automotive aerodynamics investigations carried out using computational fluid dynamics tools. The model is available in three configurations: a fastback, estate, and sedan, the latter of which is used in the present study. We generate a dataset of 1024 samples obtained by varying five parameters: the ride height, the wing-mirror lengths, the offset of the radiator grille, the length of the door handles, and the length of the boot spoiler. The ranges of these parameters are shown in Table 1. The parameters are sampled uniformly between the ranges. The ranges chosen are broader than those encountered in practice, to test performance against a wider range of drag values. Discrete modifications to distinct features such as these are the paradigm for automotive optimization, where the basic exterior shape and form is dictated by other (and sometimes conflicting) considerations such as aesthetics, regulatory compliance, and manufacturability. From the dataset, 900 samples are used to train the neural networks, 100 samples are used for validation while training, and the remaining 24 samples are used for model testing. The images are sampled at a resolution of 800×500 pixels, with each pixel binning the normal distance from a plane representing the sensor of the camera into two bytes of unsigned integer data ($2^{16} - 1 = 65535$ distinct values).

2.2. *Composite and synoptic projections*

To image the wetted surface of the automobile, one can consider, as others (Song *et al.* 2023) have done, a composite view comprising different isometric views. While this view (Figure 1) provides coverage of most of the exterior surfaces and preserves relative scaling of the features, it has the drawback of being storage-intensive owing to redundancies (for example, the wing mirrors appear in all four images) and the introduction of whitespace around each of the components of the composite image. Thus, in this work, we propose a single, synoptic view that mitigates some of these drawbacks by being uniquely suited to automotive geometries. This view, shown in Figure 2, is the result of projecting the external surface of the model onto cylindrical surfaces which are then developed into two–dimensional planes. The sequence of operations is formally invertible, resulting in a manifold exterior surface that can then be imaged. This view highlights the features of the automotive relevant for drag, while maintaining a compact representation which allows for higher-resolution sampling of features. The radius of the cylinders used for the projections are determined by the geometry of the model (the track width, front and rear

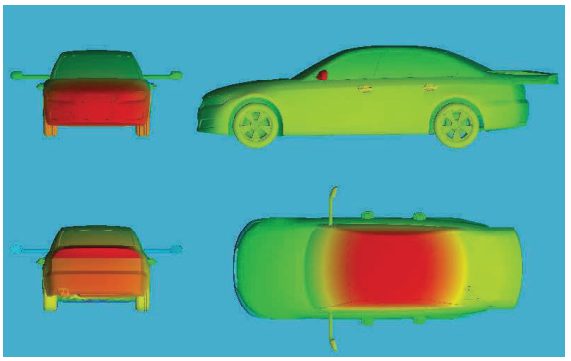
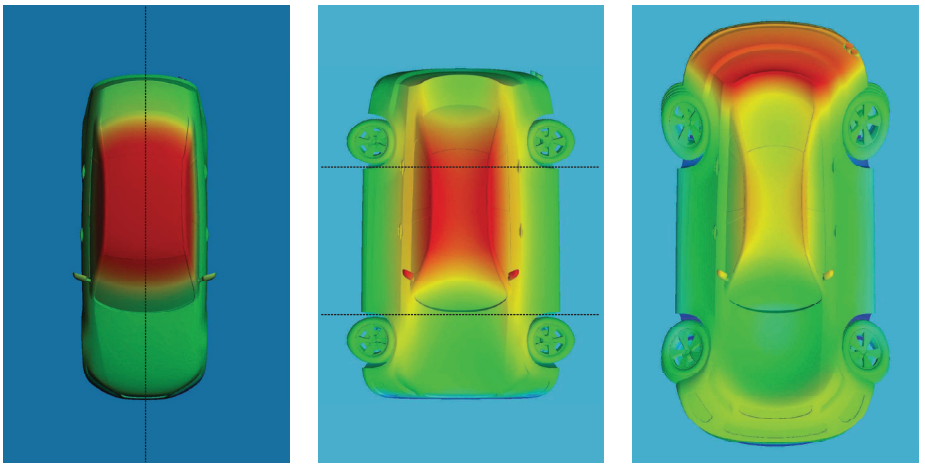


FIGURE 1. Composite view of the DrivAer geometry made up of four planar projections. The image is colored by distance normal to the camera.



(A) (B) (C)

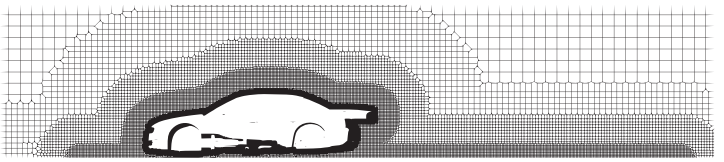
FIGURE 2. Top view of the DrivAer model showing the development of the synoptic view sequentially: (a) unmodified, (b) after projection about longitudinal axis (black line in (a)), (c) after projections about two transverse axes (dotted lines in (b)).

overhangs); in our study, these are fixed across the whole dataset. A drawback of this representation is that it introduces distortion from the polar transformation; however, we verify that this does not affect the performance in the surrogate models below.

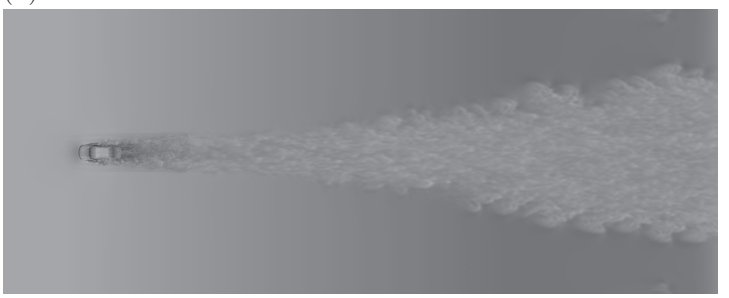
3. Surrogate modeling with CNNs

3.1. *Large-eddy simulations*

To generate the training dataset for the neural network, large-eddy simulations (LES) of the airflow past the DrivAer geometry at 85 mph with no wheel rotation are performed 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.*



(A)



(B)

FIGURE 3. (a) View of the mesh used for the large-eddy simulations, with refinement zones shown. (b) Instantaneous velocity magnitude contours showing wake behind the car.

2021). The mesh uses 6 million control volumes, with refinement near the no-slip surfaces of the car, as shown in Figure 3(a). The Vreman subgrid model is used to approximate the effects of the subgrid scales on the resolved scales. The flow is initialized with a converged initial condition from the baseline DrivAer geometry. The simulation is run for a total time of one second, and statistics are collected for the last 0.6 seconds. A sample flow visualization depicting the turbulent wake aft of the car is shown in Figure 3(b).

3.2. *CNN model evaluation*

We train on both the composite and synoptic views of the DrivAer dataset, to predict the drag coefficient. The CNNs have two convolutional layers, each with an associated pooling layer, and use ReLU activations. They have $\mathcal{O}(10^4)$ parameters. The present investigations are limited to the same Reynolds number; thus, the networks take in only geometrical representations and make predictions only for the flow conditions in the training dataset. The CNNs are trained to minimize the mean squared error (MSE) of the drag coefficient predictions for 1000 epochs. Figure 4 shows the predictions of the model on the testing set of 24 samples. It is seen that the network predicts C_D to within 2% error (with the reference being the results from the coarse LES calculations) across the whole testing set. The probability density function (PDF) of the C_D values in the training set is displayed in the same figure, and shows a correspondence between the poorly predicted samples in the training set and the sparsely sampled regions of the training data. Both the composite and synoptic projections show similar performance, showing that the synoptic view does not result in loss of information essential to predicting drag.

The training database varies all five independent parameters simultaneously; thus, to test whether the CNN has learnt to identify the contributions of individual features, we conduct a one factor at a time (OFAT) test by making predictions on six additional samples that vary only the extension of the wing mirrors (p_3 in Table 1). Figure 5 (a)

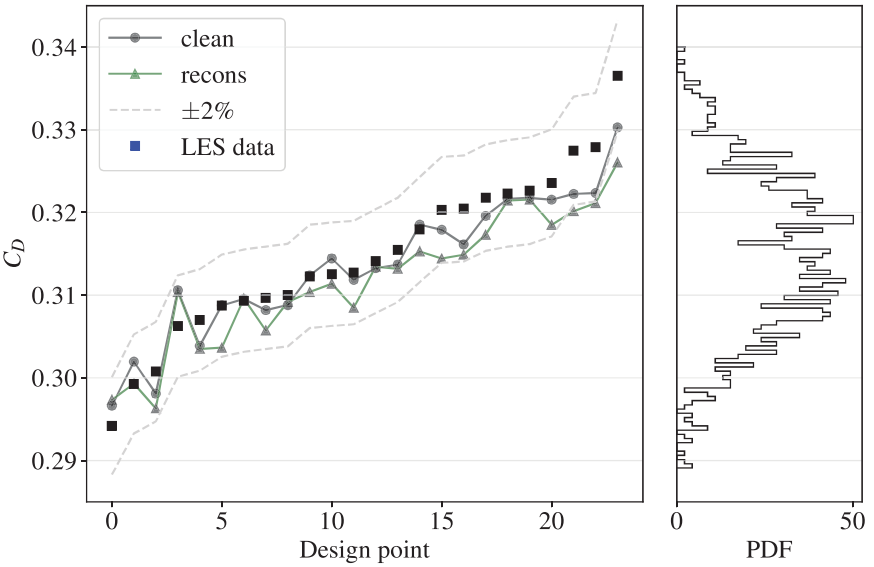


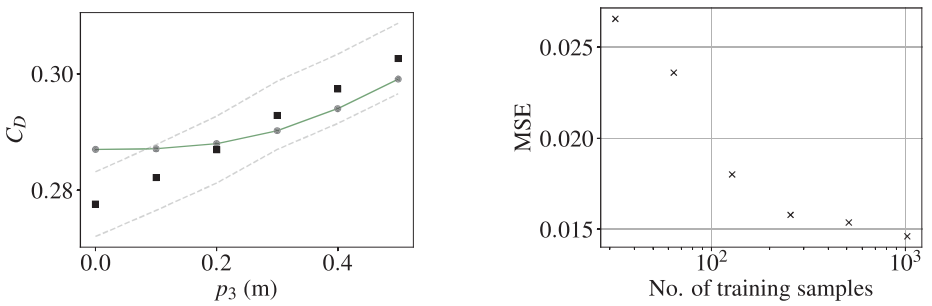
FIGURE 4. Performance of CNNs on testing set of 24 samples, arranged by increasing C_D. Squares: LES; triangles: synoptic projection; circles: composite projections; dashed lines: 2% error bounds. PDF of the C_D in the training dataset is also shown.

shows that the network has learned the drag response to the variation in this parameter; however, the performance for the first sample ($p_3 = 0$), which is not contained in the training set ranges, is poor. Our empirical observations of the relationship between C_D and p_3 suggest a locally linear manifold structure within the five-dimensional design parameter space. A data-driven model can capitalize on such low-dimensional structures, offering a computationally efficient alternative to full CFD simulations of each design point, further motivating the employment of such surrogates.

An ablation study of the training set size is conducted, the results of which are presented in Figure 5 (b), showing the effects of number of training samples on the MSE of the testing test. One observes that comparable accuracy may be achieved even with a fourfold reduction in sample count. Even with the somewhat contrived nature of the present design space, it is encouraging to see that the amount of training data required for the surrogate model is tractable.

4. Parameteric representation with CAEs

We next turn to the question of geometry parameterization. In the previous section, the generation of each sample requires the manipulation of a surface representation of the DrivAer model. While not as expensive as the LES evaluation of drag, one may consider a surrogate representation that recognizes that there are only five independent parameters in the dataset. Suppose that we were able to compress the dataset into five variables: one per parameter p_i. One would then be able to generate new designs by varying these parameters, and simply decompressing the encoded representation. Our structured image representations of the DrivAer geometry allow for the problem to be



(A) (B)

FIGURE 5. (a) OFAT performance of the CNN, varying the wing mirror lengths (p_3). Squares: LES; circles: CNN; dashed lines: 2% error bounds. (b) Ablation study of training dataset size. For both (a) and (b), the synoptic view is used.

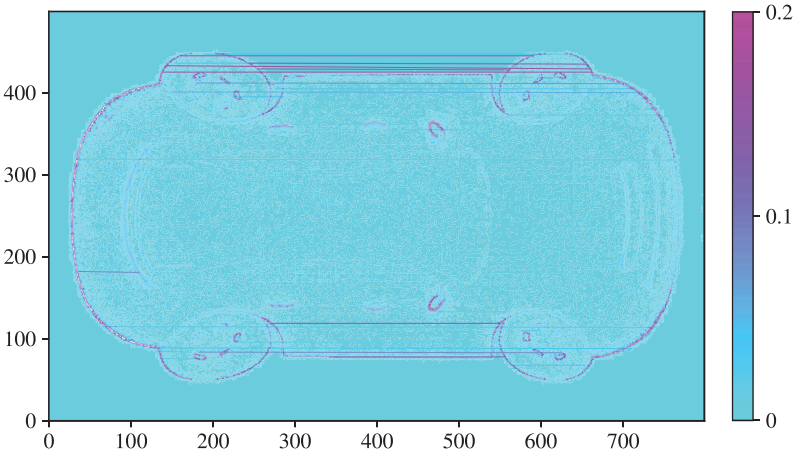


FIGURE 6. CAE reconstruction error for a training sample with parameters $p_i = \{0.00445, 0.154, 0.460, 0.0462, 0.0215\}$

.

naturally applied to a variety of linear (matrix) methods (Liberty *et al.* 2007)(Gene & Reinsch 1971); however, the complexity of the problem precludes linear models from achieving substantial compression. Thus, we use deep CAEs, which can be thought of as non–linear extensions of the proper orthogonal decomposition (Milano & Koumoutsakos 2002).

4.1. *Convolutional autoencoders*

AEs are unsupervised machine learning architectures that aim to identify low-dimensional representations of the input data. The architecture of an AE consists of two neural networks, the encoder and the decoder. The approach is unsupervised because the inputs and the outputs are typically identical: the encoder takes the input data and learns an equivalent low-dimensional representation, i.e., a vector in the latent space; the decoder

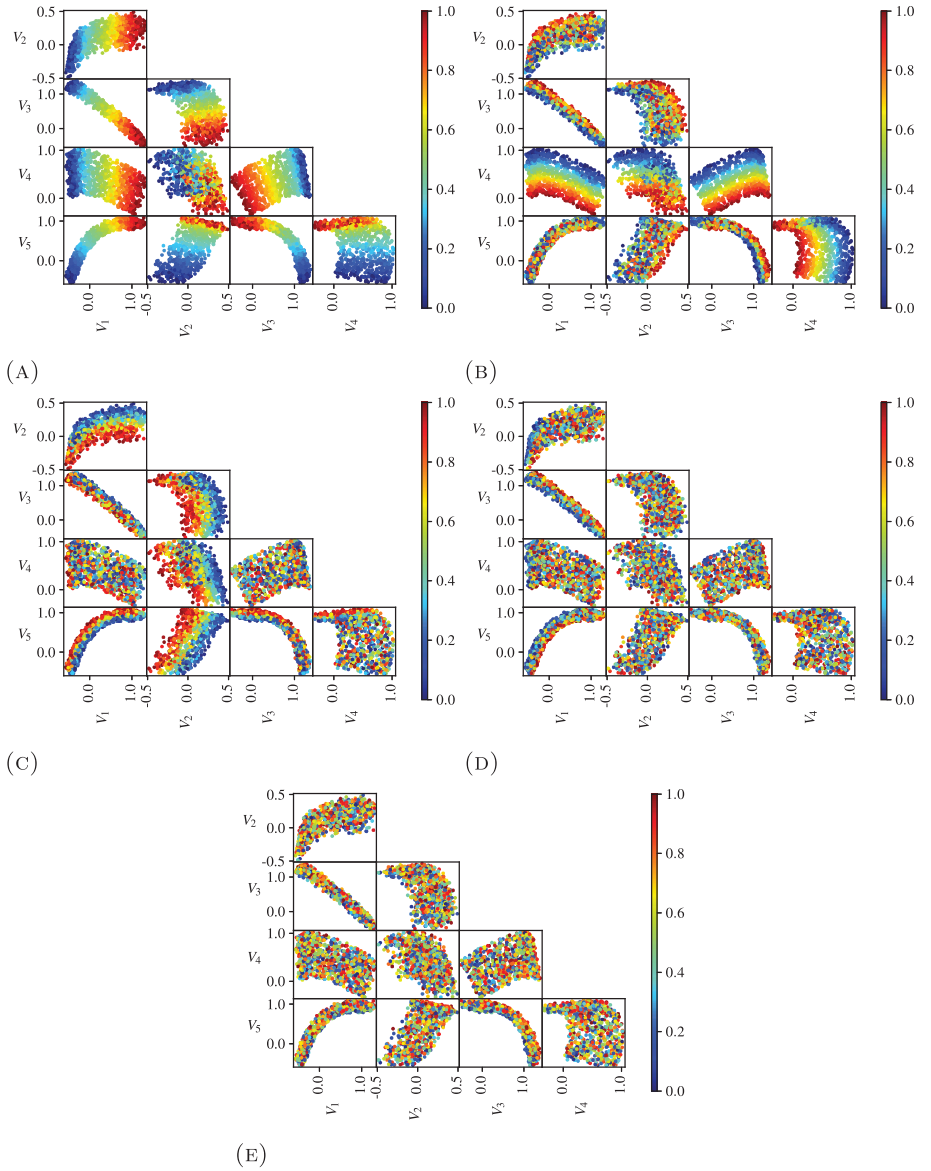


FIGURE 7. Planar projections of the latent space, colored by the database design parameters (normalized between 0 and 1) shown in Table 1: (a) ride; (b) spoiler; (c) wing mirrors; (d) radiator; (e) door handles.

starts from the latent vector and reconstructs the input. An AE can be summarized mathematically as

$$\boldsymbol{V} = e\left(\boldsymbol{x}\right); \qquad \hat{\boldsymbol{x}} = d\left(\boldsymbol{V}\right); \qquad \hat{\boldsymbol{x}} = f\left(\boldsymbol{x}\right) = d\left(e\left(\boldsymbol{x}\right)\right) \ , \tag{4.1}$$

where $e\left(\right)$ is the encoder, $d\left(\right)$ is the decoder, $\boldsymbol{V}$ is the vector of latent variables, and $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ are respectively the input and its reconstruction. $\boldsymbol{x}$, $\hat{\boldsymbol{x}} \in \mathbb{R}^n$; $\boldsymbol{V} \in \mathbb{R}^{N_{ls}}$, where n is the dimension of the input data and N_{ls} is the dimension of the latent space. The

Layer	Kernel size	Stride	Channels	In. units	Out. units	Activation
Conv.	10	5	10	-	-	ReLU
Conv.	3	2	20	-	-	ReLU
Conv.	3	2	100	-	-	ReLU
F.C.	-	-	-	93600	200	ReLU
F.C.	-	-	-	200	5	-

TABLE 2. Architecture of the convolutional encoder. For brevity, the decoder is not reported as its architecture is the mirrored encoder. Conv.: convolutional layer; F.C.: fully connected layer; In.: input; Out.: output.

architecture of the encoder adopted in this work is reported in Table 2. For brevity, the decoder is not reported since its architecture is the encoder mirrored.

The model is trained using the loss function

$$\mathcal{L} = \frac{1}{N}\sqrt{\sum_{i=1}^{N}\left[\log(x_i+1)-\log(\hat{x}_i+1)\right]^2} + \beta\frac{1}{N_e}\sqrt{\sum_{j=1}^{N_e}\boldsymbol{\ell}_{e_j}^2}, \tag{4.2}$$

where N is the number of cases; x_i and $\hat{x}_i$ are the input and its reconstruction; β is the weight of the regularization term; $\boldsymbol{\ell}_e$ are the encoder trainable parameters, $\boldsymbol{\ell}_e \in \mathbb{R}^{N_e}$. Eq. (4.2) is composed by the root mean squared error (RMSE) and a regularization term that aims at reducing overfitting; we adopted a simple L_2 norm and applied the regularization only to the encoder weights. Note that the RMSE is computed on the natural logarithm (log) of the input and its reconstruction, to emphasize small geometrical variations. The reconstruction error for a single sample is shown in Figure 6; we see a localization of the error around the varying features. Note that the change in ride height results in a rescaling of the projection in the image (since the camera is at a fixed distance from the vehicle for all samples); thus, we see a concentration of the error around the outline of the vehicle.

4.2. *CAE reconstruction evaluation*

The AE is trained to reconstruct the synoptic projections of the DrivAer geometries, using the same dataset used for training the CNNs. Given that the five design variables p_i vary independently, the dimension of the latent space has to be at least five. Saetta *et al.* (2023) showed that when an AE was trained on a database of 2D airfoil geometries in which only the thickness and camber were varying independently, the two latent variables were correlated with variations in these two parameters.

Figure 7 shows planar projections of the five-dimensional latent space. Each point represents one sample in the training set. The projections of the latent space are colored by the database design parameters, to check for correlations. Figure 7(a-c) shows an

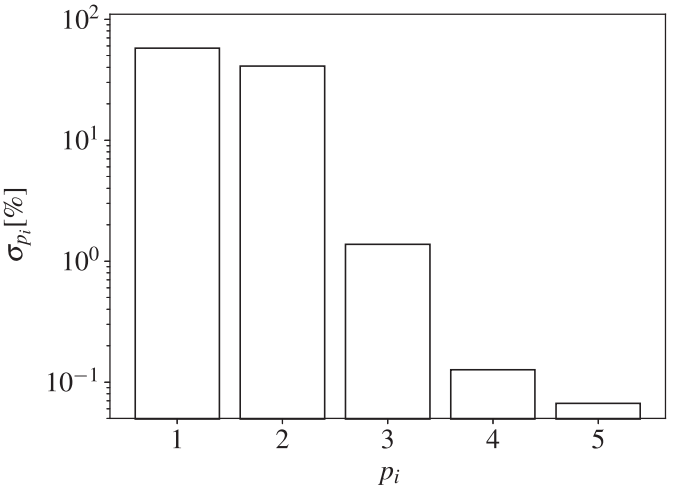


FIGURE 8. Contribution to total variance of the training dataset (σ_{p_i}) from each of the five parameters (p_i).

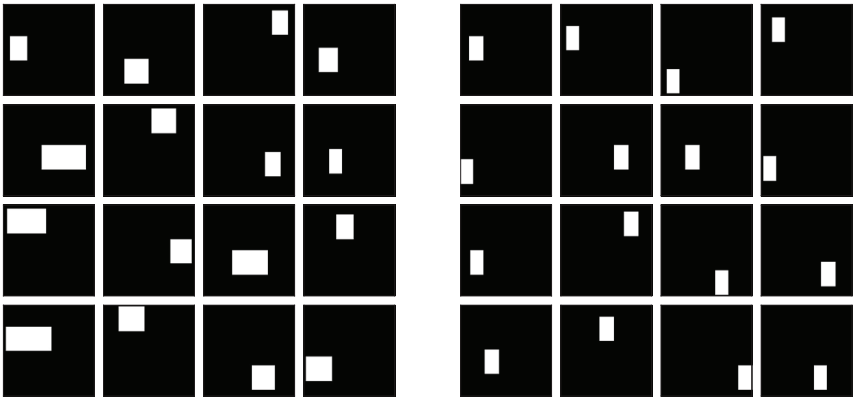
organization to the colormaps; thus, the variations in p_1 (the ride), p_2 (the spoiler), and p_3 (the wing mirrors) have been encoded in the latent space. Figure 7(d) shows a slight ordering with p_4 (the radiator) for the V_1–V_3 pair; however, the ordering is overwhelmed by noise. Figure 7(e) shows that there is no relationship with the latent variables and the remaining parameter; thus, its variation across the dataset was discarded as noise. This is owing to the fact that the total variation in p_4 and p_5 is far less than the other parameters. We quantify the contribution to total variance of the dataset from each of the five parameters as follows: given a baseline image I_0 and images I_i with the i-th parameter p_i maximized ($i = 1, \ldots, 5$), the relative impact of each parameter is quantified as

$$\sigma_{p_i} = \frac{\frac{1}{N}\sum (I_i - I_0)^2}{\sum_{j=1}^{5} \frac{1}{N}\sum (I_j - I_0)^2} \times 100\%, \tag{4.3}$$

where $\sum$ denotes the sum over all pixels and N is the total number of pixels. This equation calculates the percentage contribution of each parameter to the overall visual change in the dataset. This quantity, plotted in Figure 8, shows that variations in the first two parameters dominate the total variance in the dataset, with the third an order of magnitude smaller, and the last two another order of magnitude smaller. An increase in the number of latent variables to $N_{ls} = 10$ shows no improvement in the feature extraction capabilities of the AE. We also note that using a MSE loss function leads to a failure to detect p_3; the logarithmic MSE emphasises the smaller features, though, not sufficiently to identify p_4 and p_5.

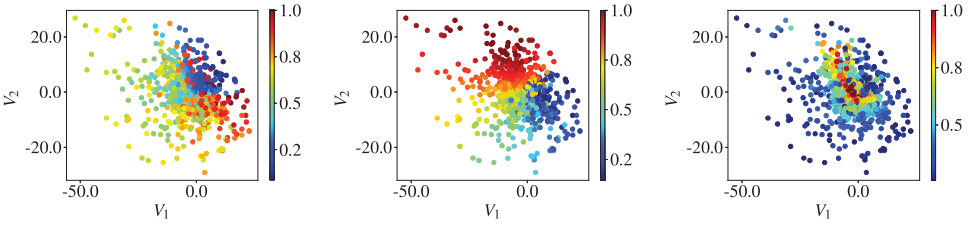
4.3. *A unit problem: rectangles*

To confirm the conjecture that the failure of the AE to encode all the independent parameters is due to their disproportionate contributions to the total variance of the dataset, we construct a unit problem that highlights this issue. The dataset consists simply of rectangular shapes that vary in location and size. Our control dataset (Figure



(A) (B)

FIGURE 9. Samples of rectangles with varying centroids and width using the (a) control dataset and (b) test dataset.



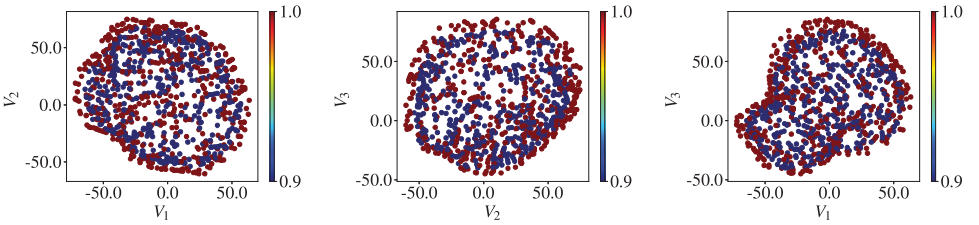
(A) (B) (C)

FIGURE 10. Projections of the latent space of the control dataset, colored by the three database design parameters: (a) x coordinate of centroid; (b) y coordinate of centroid; (c) width.

9(a)) is constructed such that all three parameters (the x and y locations of the centroid, and the width b of the rectangle) contribute in equal order of magnitude to the total variance. The test dataset differs in that the width of the rectangle changes only by one pixel, an order of magnitude smaller than the changes in the other two parameters (see Figure 9(b)). We train CAEs using these two datasets, with $N_{ls} = 3$. We find that for the control dataset, all three parameters have been encoded (see Figure 10), while for the test dataset (see Figure 11), as expected, the width of the rectangle does not correspond to an ordering in the latent space.

5. Conclusions

In this work, we develop and test two components of a data-driven framework for aerodynamic shape optimization of a realistic automotive geometry. First, we train CNNs to predict the drag coefficient from images of a projection developed to represent the wetted surface of the automotive with minimal redundancies. The networks are accurate to within 2% error. Then, we train CAEs to encode the independently varying features of the dataset, with a view to generating new geometries whose aerodynamic performance



(A) (B) (C)

FIGURE 11. Projections of the latent space of the test dataset, colored by the three database design parameters: (a) x coordinate of centroid; (b) y coordinate of centroid; (c) width.

can be evaluated by the CNNs. Prima facie investigations indicate that a representation of the geometry that highlights smaller feature changes is required to encode all the features. Interpolations within the latent space can then be used to generate new geometries.

REFERENCES

AMBO, K., NAGAOKA, H., PHILIPS, D. A., IVEY, C., BRÈS, G. A. & BOSE, S. T. 2020 Aerodynamic force prediction of the laminar to turbulent flow transition around the front bumper of the vehicle using dynamic-slip wall model LES. AIAA Paper 2018-3302.

BRÈS, G., BOSE, S., EMORY, M., HAM, F., SCHMIDT, O., RIGAS, G. & COLONIUS, T. 2018 Large-eddy simulations of co-annular turbulent jet using a Voronoi-based mesh generation framework. AIAA Paper 2018-3302.

CHANG, A. X., FUNKHOUSER, T., GUIBAS, L., HANRAHAN, P., HUANG, Q., LI, Z., SAVARESE, S., SAVVA, M., SONG, S., SU, H., XIAO, J., YI, L. & YU, F. 2015 ShapeNet: an information-rich 3D model repository. *arXiv Preprint* arXiv:1512.03012.

GENE, G. & REINSCH, C. 1971 Singular value decomposition and least squares solutions. In *Handbook for Automatic Computation: Volume II: Linear Algebra*, pp. 134–151. Springer.

GOC, K. A., LEHMKUHL, O., PARK, G. I., BOSE, S. T. & MOIN, P. 2021 Large eddy simulation of aircraft at affordable cost: a milestone in computational fluid dynamics. *Flow* **1**, E14.

HEFT, A., INDINGER, T. & ADAMS, N. 2012 Introduction of a new realistic generic car model for aerodynamic investigations. SAE Tech. Paper 2012-01-0168.

LIBERTY, E., WOOLFE, F., MARTINSSON, P.-G., ROKHLIN, V. & TYGERT, M. 2007 Randomized algorithms for the low-rank approximation of matrices. Proc. Natl. Acad. Sci. U.S.A. **104**, 20167–20172.

MILANO, M. & KOUMOUTSAKOS, P. 2002 Neural network modeling for near wall turbulent flow. *J. Comput. Phys.* **182**, 1–26.

SAETTA, E., TOGNACCINI, R. & IACCARINO, G. 2023 Machine learning to predict aerodynamic stall. *Int. J. Comput. Fluid D.* **36**, 641–654.

SONG, B., YUAN, C., PERMENTER, F., ARECHIGA, N. & AHMED, F. 2023 Surrogate

modeling of car drag coefficient with depth and normal renderings. arXiv Preprint arXiv:2306.06110.